Titre: Title:	Development of an optimization theory based tool (AutoPOM) for the Bell 427 helicopter simulation model validation
Auteur: Author:	Wen Hu Chen
Date:	2007
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Chen, W. H. (2007). Development of an optimization theory based tool (AutoPOM) for the Bell 427 helicopter simulation model validation [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7965/

UNIVERSITÉ DE MONTRÉAL

DEVELOPMENT OF AN OPTIMIZATION THEORY BASED TOOL (AutoPOM) FOR THE BELL 427 HELICOPTER SIMULATION MODEL VALIDATION

WEN HU CHEN

DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE MÉCANIQUE)
AVRIL 2007

Your file Votre référence
ISBN: 978-0-494-29220-4
Our file Notre référence
ISBN: 978-0-494-29220-4

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

DEVELOPMENT OF AN OPTIMIZATION THEORY BASED TOOL (AutoPOM)

FOR THE BELL 427 HELICOPTER SIMULATION MODEL VALIDATION

présenté par : WEN HU CHEN

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

Mme. ROSS Annie, Ph. D., présidente

M. MUREITHI Njuki W., Ph. D., directeur de recherche

Mme. BOTEZ Ruxandra, Ph. D., codirectrice de recherche

M. VO Huu-Dvc, Ph. D, membre

DEDICATION

To my wife Qing Yuan

To my son Xinyue Chen

To my parents, my sisters and brother

ACKNOWLEDGEMENTS

The work reported here was part of one project financed by the Consortium for Research and Innovation in Aerospace in Quebec (CRIAQ) and Bell Helicopter Textron Canada Limited (BHTCL), and carried out by a university team with BHTC in collaboration with the National Research Council (NRC). I wish to offer thanks to CRIAQ and BHTC for their financial support and to all those who have answered my questions during my master study.

I would like to express my sincere gratitude to the director of this thesis, Professor Njuki W. Mureithi, and the co-director, Professor Ruxandra Botez for offering me the opportunity to pursue my masters studies in such a challenging and exciting research field, and for their invaluable guidance and encouragement throughout the work involved in this thesis.

I am grateful to Mr. Adrian Hiliuta and Mr. Michel Beaulieu of LARCASE at École de Technologie Supérieure for their many very helpful technical discussions and advice.

I gratefully acknowledge the support provided by Mr. Joey Seto from the part of BHTCL for providing the data which is used in this thesis and also for his useful technical discussion and advice.

Finally my special thanks go to my wife, my son, and to my parents, my sisters and my brother for their endless support and understanding that drive me to go for success.

RÉSUMÉ

Ce mémoire fait partie d'un projet de recherche sur le simulateur de l'hélicoptère Bell 427. Le but de la recherche était d'obtenir un modèle global de simulation consistant en un ensemble de plusieurs modèles distincts de simulation correspondant à diverses conditions de vol de l'hélicoptère Bell 427. La tâche première de l'équipe était de corriger les données d'un vol d'essais et la validation du modèle de simulation.

Le but des corrections était de réorganiser les données du vol en un format standard pour que tous les associés puissent les utiliser. La validation du modèle de simulation sert à l'évaluer et approuver en accord avec « Advisory Circular AC 120-63 ». Pour valider le modèle de simulation, l'équipe a employé le logiciel POM « Proof of Match» développé par le CNRC. Habituellement, un procédé de validation d'un essai de vol prend environ de 4 à 6 heures avec POM. Dans ce projet de recherche le logiciel AutoPOM a été développé. Au moyen d'AutoPOM, un procédé de validation d'un essai de vol prend environ entre 20 et 30 minutes avec un ordinateur équipé d'un processeur AMD Athlon™ XP 1800+ (1.53GHz) et 512MB. C'est 12 fois plus rapidement que POM

Ce mémoire présente le modèle mathématique de l'hélicoptère, les corrections des données des essais de vol et la validation du modèle de simulation. La contribution de cette thèse est le développement du logiciel AutoPOM.

Le procédé de validation pour des essais de vol au moyen du logiciel POM peut être décrit comme un problème mathématique d'optimisation à multi objectifs. Ce problème peut être exprimé par des fonctions des moindres carrés des coûts avec leurs poids.

Puisque le vecteur de décision de notre problème d'optimisation est différent du vecteur d'optimisation, les expressions pour les premières dérivées partieles de la fonction objective en ce qui concerne le vecteur de décision ne peuvent pas être trouvées. Ainsi le problème d'optimisation ne peut pas être résolu par la méthode indirecte, mais il peut l'être fait par la méthode directe et l'algorithme génétique. Les trois algorithmes suivant ont été choisis : méthode de Hooke et Jeeves, méthode de Nelder et Mead, et algorithme génétique. Ces algorithmes ont été appliqués au logiciel POM et le logiciel AutoPOM a été développé. L'interface d'AutoPOM a été créée par Matlab (technique graphique de l'interface GUI), qui est clair et facile d'utilisation. C'est un critère important pour l'ingénieur dans l'industrie.

Les résultats expérimentaux prouvent que les trois algorithmes peuvent résoudre le problème d'optimisation. La méthode de Hooke et Jeeves commence à partir d'un point; ses étapes d'exploration d'échec retardent la vitesse de convergence. La méthode de Nelder et Mead commence à partir de (n+1) points; elle emploie plus d'information des (n+1) points et chaque étape rapproche de la convergence. La vitesse de convergence du l'algorithme génétique (GA) change en raison du choix aléatoire des emplacements de croisement et les emplacements de mutation; la progéniture peut être pire que ses parents et ainsi la vitesse de convergence peut être rapide ou lente.

En comparant les vitesses de convergence des trois algorithmes, nous recommandons la méthode de Nelder et Mead.

ABSTRACT

This thesis presented here is a part of work for Bell 427 helicopter simulator research project. The purpose of the research was to obtain a global simulation model which was a composition of several distinct simulation models of the Bell 427 helicopter in various flight conditions. The main tasks of our university team were the corrections of the raw flight test data and the simulation model validation.

The corrections of the raw flight test data were to reorganize the flight test data in standardized formats so that all partners could easily use it. The validation was to evaluate and approve the global simulation model in accordance with Advisory Circular AC 120-63. Our team used the software POM developed by NRC to validate the simulation model. Usually, one flight case validation process took about 4-6 hours or more by use of POM. In this research project, the software AutoPOM based on POM was developed. By use of AutoPOM, one flight case validation process took about 20-30 minutes in the computer (AMD Athlon™ XP 1800+, 1.53GHz, 512MB). This was 12 times faster than POM.

This thesis presents the helicopter mathematical model, corrections of the raw flight test data and the simulation model validation. The contribution of this thesis was the development of the software AutoPOM.

Validation process for flight cases by use of POM can be described as a mathematical multi-objective optimization problem. This problem can be expressed by a utility function of Least Square cost functions with their weights. Because the decision vector is different from the optimization vector, the expressions for the first partial derivatives

of the objective function with respect to the decision vector can not be found. Thus the optimization problem can not be solved by indirect method, but it can be by direct methods and genetic algorithms. Three algorithms (Hooke and Jeeves' method, Nelder and Mead's method, and Genetic algorithm) were selected and applied to the software POM to develop the software AutoPOM. The AutoPOM interface is created by use of Matlab Graphical User Interface (GUI) techniques, which is clear and easy to use. This is an important criterion for engineers in industry.

Experimental results show that the three algorithms can solve the optimization problem. The Hooke and Jeeves' method starts from one point, its failure exploration steps delay the convergence speed. The Nelder and Mead's method starts from $(n+1)$ points; it uses more information from $(n+1)$ points and each step is forward to the convergence zoom. The Genetic Algorithm (GA) convergence speed varies due to random choice of the crossover sites and the mutation sites, the offspring may be worse than their parents, the convergence speed can be either fast or slow.

Based on the convergence speeds of the three algorithms, we recommend the use of the Nelder and Mead's method.

CONDENSÉ EN FRANÇAIS

DÉVELOPPEMENT D'UN OUTIL BASÉ SUR UNE THÉORIE D'OPTIMISATION (AutoPOM) POUR LA VALIDATION DU MODÈLE DE SIMULATION DE L'HÉLICOPTÈRE BELL 427

01. INTRODUCTION

Ce projet a été financé par le consortium de recherche et d'innovation aérospatiale au Québec (CRIAQ) et par « Bell Helicopter Textron Canada Limited » (BHTCL). Il a été réalisé par notre équipe interuniversitaire, avec l'aide de BHTCL et le CNRC. L'objectif du projet était d'obtenir un modèle global de simulation qui consiste de plusieurs modèles distincts représentant diverses conditions de vol de l'hélicoptère Bell 427.

Dans l'industrie aéronautique, les modèles de simulation dynamique du vol sont employés pour analyser la dynamique du vol, déterminer la loi des commandes, prédire les paramètres de qualité de vol, analyser les cas d'échec, développer des simulateurs d'entraînement, et ainsi de suite.

Dans ce projet, BHTCL a effectué des essais de vol et a construit une vaste base de données pour l'hélicoptère Bell 427. Cette base de données a été divisée en deux parties. La première partie a été employée par le CNRC pour l'identification du modèle de simulation; la deuxième partie a été employée par l'équipe interuniversitaire pour la validation du modèle de simulation. L'équipe a validé (c.-à-d. évalué et approuvé) le modèle global en accord avec ce qui est requis pour un simulateur de niveau D, le plus haut niveau de fidélité pour un simulateur. Les prérequis pour le niveau D sont décrit par

le «Advisory Circular AC 120-63 » [15] au moyen du logiciel POM développé par le CNRC.

La contribution de cette thèse est principalement le développement du logiciel AutoPOM basé sur le POM de le CNRC. AutoPOM est une combinaison de POM, d'algorithmes et de techniques Matlab Interface Graphique Utilisateur (GUI).

02. ÉQUATIONS DU MOUVEMENT DE L'HELICOPTERE

Les équations de mouvement d'un hélicoptère sont démontrées en utilisant les lois de mouvement de Newton. Afin de démontre les équations de mouvement d'un hélicoptère (ou de la dynamique de l'hélicoptère), il est nécessaire de faire quelques suppositions:

a. La masse de l'hélicoptère reste constante pendant toute l'analyse dynamique. Au fait, il y une différence considérable entre la masse d'un hélicoptère avec et sans carburant, mais la quantité de combustible consommé pendant la durée de l'analyse dynamique peut être négligée sans risque.

b. L'hélicoptère est un corps rigide.

c. La terre est une référence d'inertielle.

Selon ces suppositions et en appliquant les lois de mouvement de Newton, les équations du mouvement à 6 degrés de liberté de l'hélicoptère sont exprimées comme suit:

$$\left.\begin{aligned} \dot{u} &= \tilde{X} - (wq - vr) - g\sin\theta \\ \dot{v} &= \tilde{Y} - (ur - wp) + g\cos\theta\sin\phi \\ \dot{w} &= \tilde{Z} - (vp - uq) + g\cos\theta\cos\phi \end{aligned}\right\} \quad \text{sont les vitesses de translation.}$$

$$\left.\begin{aligned} \dot{p} &= \frac{1}{I_x}L + \frac{(I_y - I_z)}{I_x}qr + \frac{I_{zx}}{I_x}(\dot{r} + pq) \\ \dot{q} &= \frac{1}{I_y}M + \frac{(I_z - I_x)}{I_y}pr + \frac{I_{zx}}{I_y}(r^2 - p^2) \\ \dot{r} &= \frac{1}{I_z}N + \frac{(I_x - I_y)}{I_z}pq + \frac{I_{zx}}{I_z}(\dot{p} - qr) \end{aligned}\right\} \quad \text{sont les vitesses angulaires de rotation}$$

Les angles d'Euler (ϕ, θ, ψ) représentent l'attitude d'un hélicoptère par rapport à la terre. La relation entre (ϕ, θ, ψ) et les vitesses angulaires sont exprimées comme suit:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Les équations ci-dessus sont non-linéaires, alors la théorie de perturbation est utilisée pour linéariser les équations et ainsi obtenir un model de simulation linéaire:

$$\left.\begin{aligned} \dot{\bar{x}}(t) &= A\bar{x}(t) + B\bar{u}(t) + \bar{f}(t) \\ \bar{y}(t) &= C\bar{x}(t) + D\bar{u}(t) \end{aligned}\right\}$$

Les matrices A and B contiennent les paramètres inconnus représentant les dérivée de la stabilité et du contrôle. En premier, les paramètres inconnus doivent être identifiés,

Ensuite il faut évaluer et valider les paramètres identifiés, en s'assurant que ceci est fait sur des donnés du même essai de vol et avec les mêmes conditions de vol.

03. ESSAIS EN VOL ET LES CORRECTION DES DONNÉES

Une tâche de l'équipe interuniversitaire était de corriger les donnés des essais de vol. Les de vol servent à recueillir des données et d'établir une base de données des caractéristiques de vol d'un hélicoptère et de ses sous-systèmes pour les analyser au sol. Les données enregistrées doivent être prétraitées, corrigées et organisées dans des formats normalisés de sorte que tous les partis impliqués puissent les utiliser.

Les instruments pour les essais de vol et les paramètres enregistrés sont les suivants :

(1) Devant l'hélicoptère, une perche de nez avec des sondes de pression et des ailettes est montée à l'extérieur pour éviter des interactions avec le rotor. Elle est employée pour mesurer la vitesse totale de l'air, l'angle d'attaque et l'angle de dérapage (V, α, β); à partir des ces mesures, les composantes u, v, w de la vitesse, peuvent être calculées.

(2) Un gyroscope mesure la vitesse de roulis (p), tangage (q) et lacet (r). alors qu'un compas gyroscopique mesure des angles de roulis (ϕ), de tangage (θ) et de lacet (ψ). Des accéléromètres mesurent les accélérations longitudinales, latérales et verticales (a_x, a_y, a_z).

(3) Un capteur de pression mesure l'altitude et le taux de montée;

(4) Des potentiomètres aux commandes du pilote (bâton, pédales, collectifs) mesurent les entrées de contrôle (δ_{long}, δ_{lat}, δ_{ped}, δ_{col}) ;

(5) Un capteur de température mesure la température extérieure (OAT).

Les raisons pour corriger les données sont les suivantes :

(1) les instruments peuvent présenter des erreurs entre les vraies données et les données enregistrées;

(2) le temps d'enregistrement des instruments peut induire des erreurs de temps dans les paramètres mesurés;

(3) certaines données doivent être présentées dans les unités appropriées pour les utiliser aux États-Unis, telles que le carburant en livre, les vitesses en noeud, altitudes en pied, etc.

(4) certaines données doivent être transférées du centre d'instrumentation (IC) situé au centre de gravité (CG), parce que l'IC et certaines sondes telles que celles sur la perche de nez, etc. sont situés à différents points par rapport au CG.

04. VALIDATION DU MODÈLE DE SIMULATION

L'hélicoptère est contrôlé par les équations dynamiques suivantes:

$$\dot{\bar{x}}(t) = A\bar{x}(t) + B\bar{u}(t) + \bar{f}(t)$$

où, $\bar{x}(t) = [u, w, q, \theta, v, p, \phi, r]^T$, $\bar{f}(t)$ est le vecteur des perturbations atmosphérique

La solution du modèle de simulation peut être écrite comme suit:

$$\bar{x}(t) = \bar{x}_0 + \int_{t_0} \left(A\bar{x}(t) + B\bar{u}(t) \right) dt$$

où, $\overline{x}_0 = \overline{x}(t_0)$ est le vecteur de condition initiale au temps t_0. Ainsi, $\overline{x}_0$ affecte $\overline{x}(t)$.

Si les dérivées de stabilité dans la matrice A et les dérivées de commande dans la matrice B sont correctement évaluées à partir des données de vol par la méthode d'identification, la solution $\overline{x}(t)$ du modèle de simulation va correspondre aux données enregistrées avec les mêmes conditions de vol. Lorsque $\overline{x}(t_0)$ est correctement donné, en considérant le bruit des signaux et des perturbations inattendues, l'erreur entre les valeurs d'intégration $\overline{x}(t)$ et les essais de vol devraient être en dessous des tolérances indiquées dans Advisory le «Advisory Circular AC 120-63 » [15]. En d'autres termes, si les matrices A et B sont correctes, nous trouverons un $\overline{x}(t_0)$ raisonnable pour comparer $\overline{x}(t)$ avec les données de vol, et l'erreur devrait être en de ça des tolérances.

Pour la validation du modèle de simulation par rapport au données d'essais en vol, le procédé de validation doit vérifier si les graphiques de résultats de simulation sont en de ça des tolérances en changeant les composantes du vecteur de condition initiale $\overline{x}(t_0)$; si oui, le modèle est évalué « vrai », autrement le modèle est évalué « faux » et il a doit être corrigé. Pour la certification de simulateur du l'hélicoptère Bell 427, le logiciel POM est employé pour valider le modèle de simulation, qui doit répondre aux exigences les plus sévères (niveau D) de la référence [15].

Le changement d'un paramètre initial peut causer les changements d'un ou plusieurs paramètres découlant de l'accouplement entre les paramètres de l'hélicoptère. Le procédé de validation de POM est un processus d'essais et erreur qui prend beaucoup de temps. Habituellement, la validation d'un cas par essai de vol prend de 4 à 6 heures et pour cette

raison, il a été décidé de développer un logiciel (AutoPOM) pour automatiser les tâches de l'utilisateur.

05. DESCRIPTION DE PROBLÈME D'OPTIMISATION

Le procédé de validation consiste à trouver le minimum entre les données des essais de vol et les résultats de la simulation, ceci en changeant le conditions initiales. Il peut aussi être décrit comme un problème d'optimisation à objectifs multiples dont la forme discrète est la suivante:

Fonction objective (en forme discrète)

$$\min_{x\in C} F(\bar{x}) = \min_{x\in C} \sum_{i=1}^{n} w_i f_i(\bar{x}) = \min_{x\in C} \sum_{i=1}^{n}\sum_{k=1}^{m} w_i [x_{i,\,mesuré\ (t_0+k\Delta t)} - x_{i,\,simulé\ (t_0+k\Delta t)}]^2$$

qui est soumise aux contraintes:

$$C = \left\{ h^-_{i,(t_0+k\Delta t)} \le x_{i,\,simulé,\,(t_0+k\Delta t)} \le h^+_{i,(t_0+k\Delta t)} \quad i = 1,2,...n, \quad \dot{\bar{x}} = A\bar{x} + B\bar{u} \right\}$$

où, w_i sont les poids, t_0 est le temps initial, et $\bar{x}$ est le vecteur de la fonction objective.

$\Delta t = (t_{end} - t_0)/m$, Δt est l'intervalle discret de temps, m est un nombre entier qui est entré dans le code par l'utilisateur du logiciel.

$[h_i^-, h_i^+]$ sont les tolérances de $x_{i,simulé\ (t_0+k\Delta t)}$ au temps $t = (t_0 + k\Delta t)$.

$\dot{\bar{x}} = A\bar{x} + B\bar{u}$ est le modèle de simulation de la dynamique d'hélicoptère.

$f_i(x_{i,(t_0+k\Delta t)}) = [x_{i,mesuré\ (t_0+k\Delta t)} - x_{i,simulé\ (t_0+k\Delta t)}]^2$,

$\bar{x} = [x_1, x_2, ...x_n]^T$, $\bar{x}_i = x_i(\bar{y}, t)$ $i = 1,2,...,n$,

$\overline{y} = [y_1, y_2, ... y_j]$, où $\overline{y}$ est un vecteur des variables de décision, et j est le nombre de variable de décision.

La fonction objective, à l'exception des poids, comprend des unités et ordres de grandeur divers. Ainsi, des unités non dimensionnelles sont utilisées pour les pondérations. Dans le but de simplifier, l'expression de pondérations suivante est utilisée:

$$w = \frac{100}{mC_h^2}$$

où, $C_h = h^+ - h^-$, C_h est une constante qui est déterminée par le « Advisory Circular AC 120-63 ».

06. ALGORITHME D'OPTIMISATION

Les algorithmes conventionnels d'optimisation sont donnés par des méthodes de recherche indirectes et des méthodes de recherche directe. Les méthodes de recherche indirectes sont souvent employées dans le cas où le gradient peut être trouvé facilement et la fonction objective est continue et différentiable dans le domaine d'optimisation. Les méthodes de recherche directes sont employées pour résoudre les problèmes d'optimisation qui n'exigent aucune information sur le gradient de la fonction objective. Elles exigent seulement que les fonctions objectives peuvent être représentées par une formule ou des séries de formules, même une table de données. Serait suffisante. Les fonctions objectives n'ont pas besoin d'être lisses ou même continues, elles doivent seulement être des fonctions unimodales dans le domaine d'optimisation.

L'algorithme génétique (GA), proposé par John Hollande l'a proposé dans les années 60, a été largement répandu dans le domaine de l'optimisation. Ces algorithmes également n'exigent pas d'information dérivée. Les GA peuvent résoudre des problèmes que les méthodes directes conventionnelles ne peuvent pas résoudre comme les problèmes multi extremum [17].

Les caractéristiques de notre problème d'optimisation sont les suivantes:

(1) Le vecteur de décision diffère du vecteur de fonction objective. Les deux vecteurs sont liés par le modèle de simulation, mais nous ne connaissons pas le modèle de simulation en raisons des règles de confidentialité.

(2) Les expressions pour les premières dérivés partielles de la fonction objective par rapport au vecteur de décision ne sont pas connues. Seulement les résultats numériques des mesures et simulations sont disponibles.

(3) La fonction objective est la somme des fonctions unímodales.

Le problème d'optimisation, il peut être résolu au moyen des méthodes de recherche directes et GA. Dans cette thèse, nous avons choisi la méthode de Hooke et Jeeves, la méthode de Helder et Mead et l'algorithme génétique (GA). Nous avons lié les trois algorithmes à POM et avons développé le logiciel AutoPOM.

07. APPLICATION DES TROIS ALGORITHMES

Les trois algorithmes ont été liés à POM et le logiciel optimisé AutoPOM a été développé. Les vitesses de convergence de trois algorithmes sont différentes. Suite à de multiples essais au moyen d'AutoPOM, la méthode de Helder et Mead est recommandée

à cause de la rapidité de sa convergence. L'approche de Hooke et Jeeves consiste à partir d'un point et à effectuer une séquence d'étapes d'exploration dont certaines sont des étapes d'erreurs qui retardent la vitesse de convergence. De son côte, la méthode de Nelder et Mead commence la recherche à partir de $(n+1)$ points de recherche et à chaque séquence, la pire valeur est remplacée par une meilleure. Cependant, elle emploie plus d'information à chaque étape, ce qui ralentit la convergence. Enfin, la vitesse de convergence du GA varie en raison des croisements et des mutations. Les croisements et les mutations peuvent rendre la progéniture pire que leurs parents, ce qui réduit la vitesse de convergence. Par conséquent, la vitesse de convergence par le GA peut être plus rapide ou plus lente que par les autres méthodes. Par contre, la vitesse moyenne est plus lente que celle de la méthode de Nelder et Mead. Au moyen d'AutoPOM, la validation d'un essai de requiert environ entre 20 et 30 minutes. C'est 12 fois plus rapide que la validation manuelle avec POM

CONCLUSION

Le modèle de simulation d'un hélicoptère est habituellement est décrit par 6 degrés de liberté dynamiques. Les paramètres dans le modèle sont couplés. Le changement d'un des paramètres initiaux peut causer les changements d'un ou plusieurs paramètres de sortie. Ainsi, le procédé de validation du modèle de simulation prend beaucoup de temps au moyen du logiciel POM.

Le processus de validation en est un à objectifs multiples. Ce type de problème d'optimisation peut être résolu par des méthodes de recherche directe et par l'algorithme

génétique. Pour résoudre le problème d'optimisation, la méthode de Hooke et Jeeves, la méthode de Helder et Mead et la méthode et de l'algorithme génétique. ont été utilisés. La méthode de Helder et Mead a été recommandé due à sa vitesse de convergence qui est la plus rapide des trois algorithmes.

AutoPOM a été développé pour optimiser l'exécution de la validation de POM. Son interface a été créée au moyen de l'interface graphique de MATLAB (GUI) et son utilisation est simple et efficace. La validation d'un essai de vol qui prenait environ de 4 à 6 heures au moyen de POM, prend maintenant entre 20 et 30 minutes avec AutoPOM.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOTATIONS

$\{B\}$	Body-fixed coordinate.
$\{E\}$	Inertial or earth reference frame.
$F(\overline{x})$	Objective function.
$\overline{F}$	External applied force vector.
F_x	Component of the external applied forces along the longitudinal body axis.
F_y	Component of the external applied forces along the lateral body axis.
F_z	Component of the external applied forces along the normal body axis.
$\overline{H}$	Angular momentum vector about the center of gravity.
L	Moments about the normal body axis.
M	Moments about the lateral body axis.
$\overline{M}$	External applied moment vector about the center of gravity.
Mach	Mach number.
N	Moments about the normal body axis.
ROC	Rate of climb.
U	Utility functions.
$\overline{V}$	True speed vector.
X	Aerodynamic force along the longitudinal body axis.
$\tilde{X}$	Normalized force along the longitudinal body axis.
Y	Aerodynamic force along the lateral body axis.
$\tilde{Y}$	Normalized force along the lateral body axis.
Z	Aerodynamic force along the normal body axis.
$\tilde{Z}$	Normalized force along the normal body axis.
error	Relative error.
popsize	Number of population (or population size).

p_c	Crossover rate.
p_m	Mutation rate.
p	Angular rate (or roll rate) .
q	Angular rate (or pitch rate).
r	Angular rate (or yaw rate).
v_{ideal}	Ideal value.
$v_{acceptable}$	Acceptable value.
u	Component of the air velocity along the longitudinal body axis.
v	Component of the air velocity along the lateral body axis.
w	Component of the air velocity along the normal body axis.
w_i	Weight.
x_B	Longitudinal axis on body.
y_B	Lateral axis on body.
z_B	Vertical axis on body.
$\bar{y}$	Decision vector of optimization.
$\bar{\omega}$	Angular velocity vector.
ϕ	Roll angle.
θ	Pitch angle.
ψ	Yaw angle.
α	Angle of attack.
β	Side slip angle.
σ	Standard derivation.
δ_{long}	Longitudinal.
δ_{lat}	Lateral control input.
δ_{ped}	Pedal control input.
δ_{col}	Collective control input.

ABBREVIATIONS

A/C	Aircraft.
BHTCL	Bell Helicopter Textron Canada Limited.
CRIAQ	Consortium for Research and Innovation in Aerospace in Quebec.
IAS	Indicated Air Speeds.
CG	Center of the Gravity.
DoF	Degree of Freedom.
GA	Genetic Algorithm.
GUI	Graphical User Interface.
IC	Instrumentation Center.
MOO	Multi-Objective Optimization.
MLE	Maximum Likelihood Estimation.
LS	Least Squares.
NRC	The National Research Council.
OAT	Outside Air Temperature.
POM	Proof of Match.
QTG	Quality Test Guide.
TAS	True Air Speed.
SOO	Single-Objective Optimization.

CHAPTER 1

INTRODUCTION

The present thesis research was part of a simulator certificate program for Bell 427 helicopter. The project was done by a university team, Bell Helicopter Textron Canada Limited (BHTCL) and the National Research Council Canada (NRC). The objective of the project was to obtain a global simulation model which was a composition of several distinct simulation models of the Bell 427 helicopter in various flight conditions. In the aircraft industry, flight dynamics simulation models are built for analyzing flight dynamics, designing control laws, predicting flying quality parameters, analyzing failure cases and training simulators.

The Bell 427 is currently the latest helicopter of Bell Helicopter Textron and is designed as a civil and multiple purpose light helicopter. It can be used to carry out executive, commuter transport and cargo missions. It offers 8-place seating, two individual seats in the front and two 3-place bench seats in a club configuration in the back. The Bell 427 is a twin-engine helicopter with a 4-bladed main rotor system and a 2-bladed tail-rotor system.

The Bell 427 made its maiden flight on December 11, 1997, in Quebec. It was awarded Canadian certification by Transport Canada on November 19, 1999, US certification in January 2000 and US FAA dual pilot IFR (Instrument Flight Rules) certification in May 2000. A three view drawing of the Bell 427 is given in Figure 1-1. The Bell 427's general characteristics and performance are as follows:

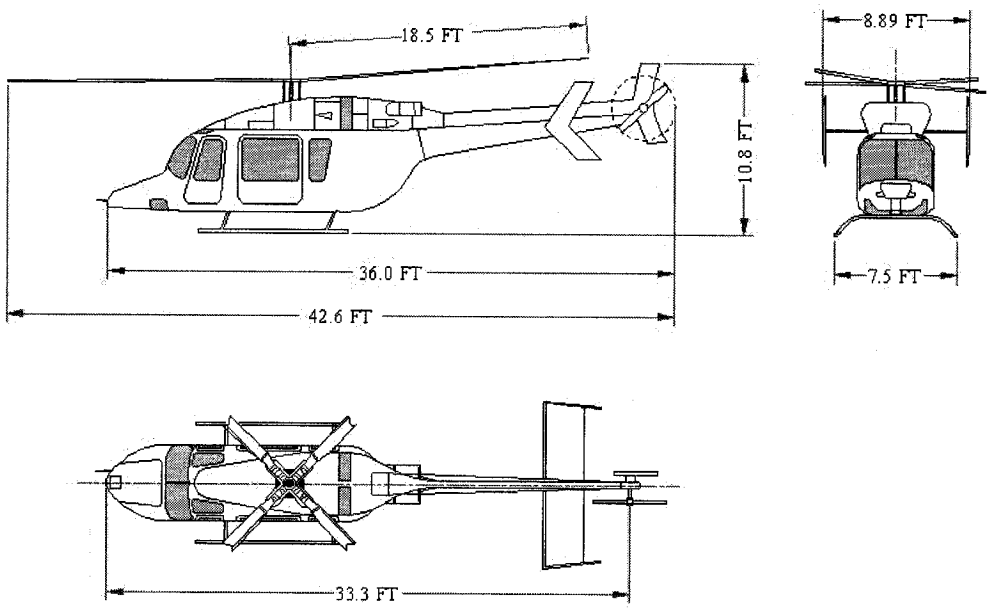


Figure 1-1 A three view drawing of the Bell 427

General characteristics:

- Powerplant: 2x Pratt & Whitney Canada PW207D, 550 hp (410 kW)
- Empty: 3875 lb (1758 kg)
- Maximum takeoff: 6550 lb (2971 kg)

Performance:

- Maximum speed: 136 knots (251 km/h)
- Range: 387 miles (716 km)
- Service ceiling: 18900 ft (5761 m)
- Rate of climb: 2,000 ft/min (609.6m/min)
- Endurance, standard fuel, no reserves : 4 hours

Like any physical system, airplanes and rotorcrafts must obey the basic laws of physics. Figure 1-2 shows the force balance of an airplane and a helicopter in forward flight. For a fixed wing airplane, its thrust is generated when its engine pushes the air backwards and its lift force is generated when its airfoil is moving in the air, i.e. an airplane has separate means of generating forces for forward propulsion and lift forces. The lift balances the weight and the thrust balances the drag. Therefore the airplane motions usually can be dealt with in two groups, the longitudinal equations and the lateral-directional equations. A helicopter, on the other hand, uses only the thrust from a rotor to meet forward propulsion force and lift force for sustained flight. The rotor thrust vector must balance not only the gross weight, but also the lift and the drag. Therefore helicopter motions usually are described by 6 DoF dynamics.

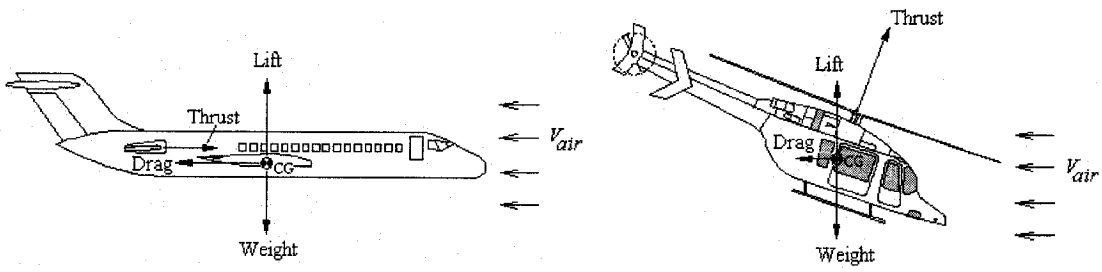


Figure 1-2 Forces on an airplane and a helicopter in level flight

In this project, BHTCL conducted flight tests and built a huge database of the Bell 427 flight tests. This database contains two parts, the first part is a set of 2311 control

input manoeuver flight tests which is used to identify stability derivatives and control derivatives*, the second part is a set of flight tests recorded in the same flight conditions as the first part which is used for the simulation model validation.

NRC developed copyrighted software for the Bell 427 referred to as POM (Proof of Match). This software was used to estimate stability and control derivatives by use of the Maximum Likelihood Estimation (MLE) method, and to simulate and validate the dynamic model of the Bell 427.

The university team used the POM software to evaluate and validate the dynamic model in accordance with Advisory Circular AC 120-63 requirements of the highest level D [15]. We evaluated the trends in the model errors and communicated our observation to NRC. NRC corrected the dynamic model according to our recommendation.

The project lasted approximately two and one half years and ended at the end of July, 2006. During the first year, the main tasks were to format and correct the flight data and complete the POM software. During the rest of the time, the job was to validate the dynamics model. Almost 600 flight test cases had to be analyzed by use of the software POM. The university team contributed mainly in the raw data corrections and validation process.

The raw data corrections were to correct the instrument offset, the time delays of instrumentations and transfer accelerations, velocities from the Instrumentation Center (IC) to the Center of Gravity (CG). The corrected data were organized in standardized formats so that they can be easily used by all partners of the project.

---

* See definition page 28 (also Table 2.1 page 30)

The validation process was a trial and error process by use of POM. This was the most labour intensive and time consuming part of the project. The detailed validation process will be introduced in Chapter 4. Usually, the validation process of a flight case takes a POM operator 4-6 hours maybe more by use of the POM software. For this reason, the need to develop an automatic POM software (called AutoPOM) to carry out the POM operator's manual work became evident.

The contribution of this thesis was the development of the software AutoPOM based on the NRC's POM. The validation process is to find the minimum differences between the results obtained by simulation and the recorded flight data by changing the initial conditions. Therefore, this process can be considered as a multi-objective optimization problem with the constraints of the helicopter dynamics. This problem is to find the optimization decision vector (initial condition vector) while the multi-objective function is minimum. It can be solved by algorithms. AutoPOM is a combination of POM, algorithms and Matlab Graphical User Interface (GUI) techniques.

In Chapter 2, we describe the coordinate systems and the Euler angles, derive the helicopter 6 DoF dynamics that are used for analysis and simulation purposes, and present the linearization of helicopter dynamics as well as the expressions of state-space. The state space contains the stability and control derivatives which are the primary unknowns to be determined via experimental flight tests and system identification.

In Chapter 3, we introduce flight test, instrumentations and measurement of flight data. The recorded data are preprocessed, corrected and organized in standardized

formats so that all partners involved in the project can easily use them. We derive the transformation expressions of accelerations and airspeed from the Instrumentation Center (IC) to the Center of Gravity (CG).

In Chapter 4, we present the validation requirements according to Advisory Circular AC 120-63 requirements and the Quality Test Guide, introduce the simulation model validation process by use of software POM, and explain the time consuming nature of validation as it is currently done.

In Chapter 5, we formulate the validation process as a multi-objective optimization problem. We then focus on solving the problem. We discuss the objective function, the weights of the multi-objective function, and derive the objective function of the optimization problem.

In Chapter 6, the characteristic of the optimization problem is analyzed followed by a discussion on the solution of the problem. In this problem, because the decision vector differs from the objective function vector, we can not find the expressions for the first partial derivatives of the objective function with respect to the decision vector. This means that the problem cannot be solved by use of direct search methods. Indirect search methods and genetic algorithm (GA) are therefore proposed and elaborated. Specifically, three algorithms, Hooke and Jeeves' method (pattern search), Nelder and Mead's method, and a genetic algorithm (GA) are presented in detail.

In Chapter 7, we present the application of the three algorithms to the Bell 427 validation problem. We apply the three algorithms to the corresponding optimization problem, and develop the software AutoPOM based on POM. AutoPOM runs in Matlab

environment as POM does. Its interface is created by use of Matlab Graphical User Interface (GUI) techniques. This interface is designed to be clear and user friendly so that an AutoPOM user operates it. An important problem of the original POM which is addressed in AutoPOM is how to eliminate the trial and error decision making process of the POM operator which was necessary to 'guide' POM towards the correct solution. The design is such that the AutoPOM user can easily choose anyone of the three optimization algorithms. All of the three algorithms are proved by flight cases. The convergence speeds of the three algorithms are also tested. The software AutoPOM is attached in the CD as an appendix.

CHAPTER 2

THE HELICOPTER EQUATIONS OF MOTION

The helicopter equations of motion are derived by applying Newton's law of motion, which relates the summation of the external forces and moments to the linear and angular accelerations of the system or body. In this case, certain assumptions must be made and axis systems must be defined. In this chapter, we describe the coordinate systems, the Euler angles and the rotation matrices from one coordinate system to another. We also derive the helicopter dynamics model that will be used for analysis and simulation purposes.

2.1 Assumptions and definitions

2.1.1 Assumptions

In order to derive the equations of helicopter motion (or helicopter dynamics), it is necessary to make the following assumptions [1, 2]:

a. The mass of the helicopter remains constant during any particular dynamic analysis. (In reality, there is considerable difference in the mass of a helicopter with and without fuel, but the amount of fuel consumed during the period of the dynamic analysis may be safely neglected).

b. The helicopter is a rigid body.

c. The earth is an inertial reference.

2.1.2 Definitions of the coordinate systems

2.1.2.1 Body-fixed axis

The body coordinate system $\{B\}$ is connected to the helicopter. In general, the body axis system is fixed to the helicopter. The origin O of the body-fixed axis system is usually chosen to coincide with the center of the gravity (CG) of the helicopter (i.e. $CG \equiv O$). Figure 2-1 shows the body axes. Here x_B, y_B, z_B are the orthonormal axes. The x_B axis is the longitudinal axis and positive out the nose of the helicopter. The y_B axis is the lateral axis and positive out the right side as you sit in the cockpit and face out the front of the helicopter. The z_B axis is the vertical axis and positive normal to the x and y axis and points vertically downward when the helicopter is in level

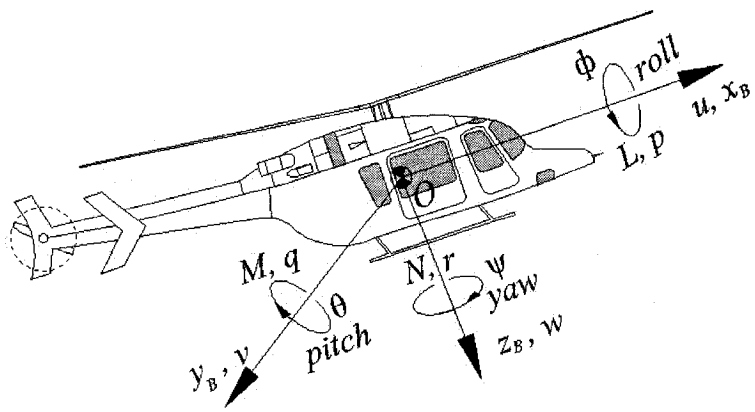


Figure 2-1 Body axis system

flight.

The speeds of *u, v,* and *w* are the linear speeds. The moment axes obey the right hand rule about each axis. Moments about the x_B, y_B and z_B axes are labeled M, L, and N, respectively. The body angular rates (*p, q,* and *r*) and Euler angles (ϕ, θ, and ψ) are also measured positively using the right-hand rule about each axis. Euler angles will be introduced in Section 2.2.

2.1.2.2 The inertial axis system

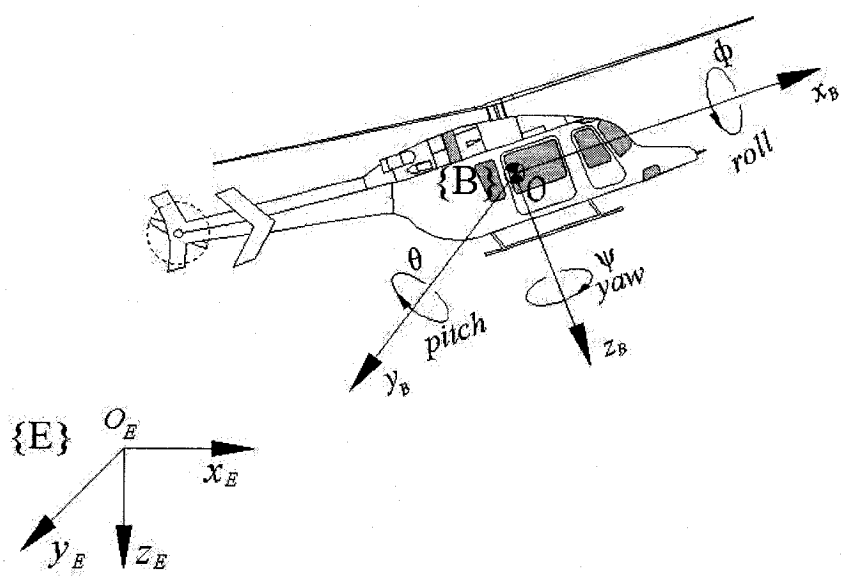


Figure 2-2 Coordinate systems

In order to describe the position, orientation and motion of the helicopter, it is necessary to define an inertial reference frame; this inertial axis system is placed at a

fixed place on the earth, so we call the inertial axis system $\{E\}$, and is defined as follows:

a. Three axes x_E, y_E, z_E are orthonormal.

b. The axes x_E and y_E form a horizontal plane

c. The axis z_E is in the direction of the gravitational field, i.e. x_E and z_E form a vertical plane. Figure 2-2 shows the two coordinate systems $\{B\}$ and $\{E\}$.

2.2 Euler angles and transformation matrices

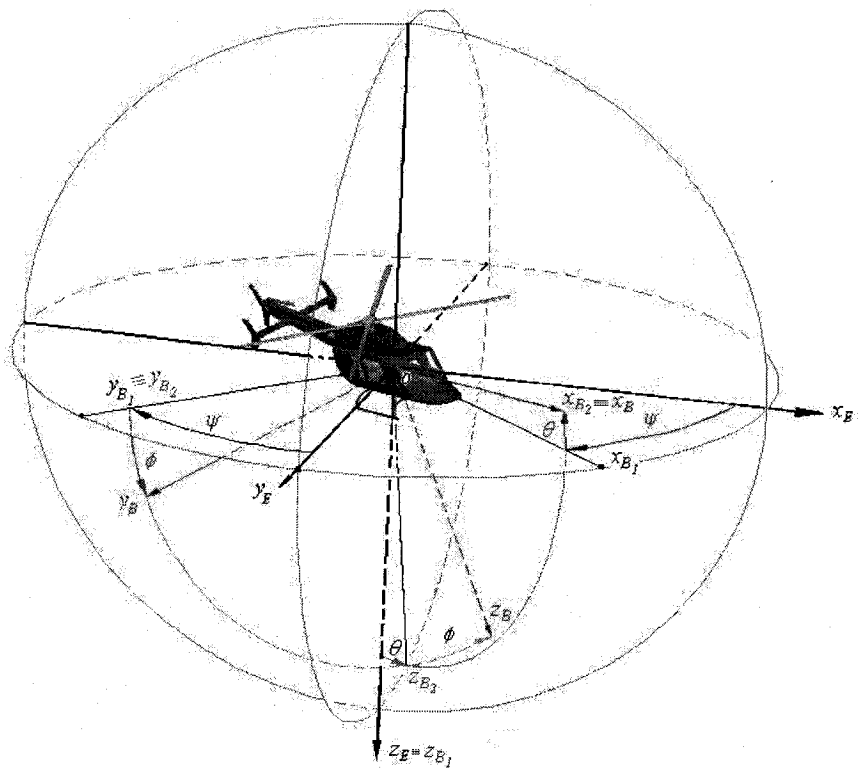


Figure 2-3 Helicopter attitude and Euler angles

Euler angles are used to describe rotations or relative orientations of orthogonal

coordinate systems. They can represent the attitude of the helicopter with respect to the earth. Their definition is not unique and there are many different conventions [3, 4, 5]. Among the different Euler angle conventions, *Z-Y-X* Euler angles are the most commonly used [6]. These angles are roll angle ϕ, pitch angle θ and yaw angle ψ. Figure 2-3 illustrates the attitude of a helicopter and Euler angles, and represents arbitrary rotations as a composition of three successive principal rotations.

The Euler angles (ϕ,θ,ψ) relate two orthogonal coordinate systems having a common origin. The transition from one coordinate system to the other is achieved by a series of two-dimensional rotations. All rotations are in a counter-clockwise sense (right-handed, mathematically positive sense). The rotations are performed about coordinate system axes generated by the previous rotation step. Figure 2-4 illustrates the rotation steps.

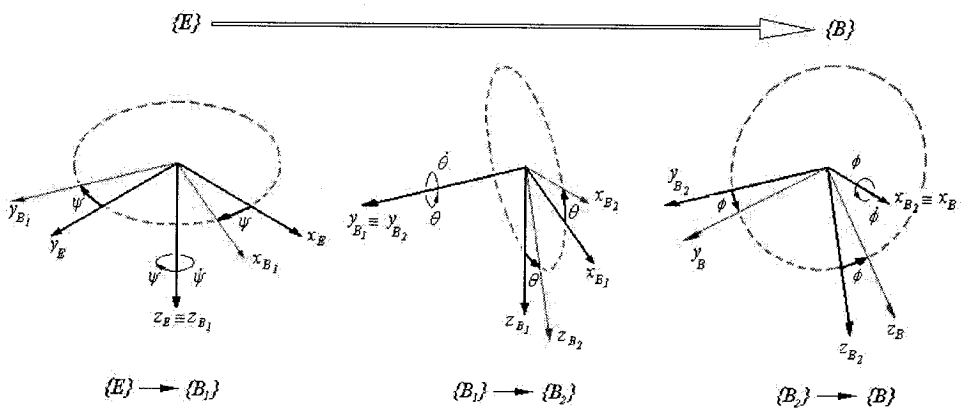


Figure 2-4 Rotations and Euler angles

In order to obtain the rotation matrix from frame $\{B\}$ to frame $\{E\}$, we apply the three rotations to a helicopter in Figure 2-5. We start by parallel translating the inertial frame

$\{E\}$ until its origin coincides with the origin of the body-fixed coordinate system $\{B\}$. We can do so because the relative displacement of $\{B\}$ and $\{E\}$ is irrelevant to describing orientation, then we rotate the inertial frame three times until it coincides with the body-fixed coordinate system.

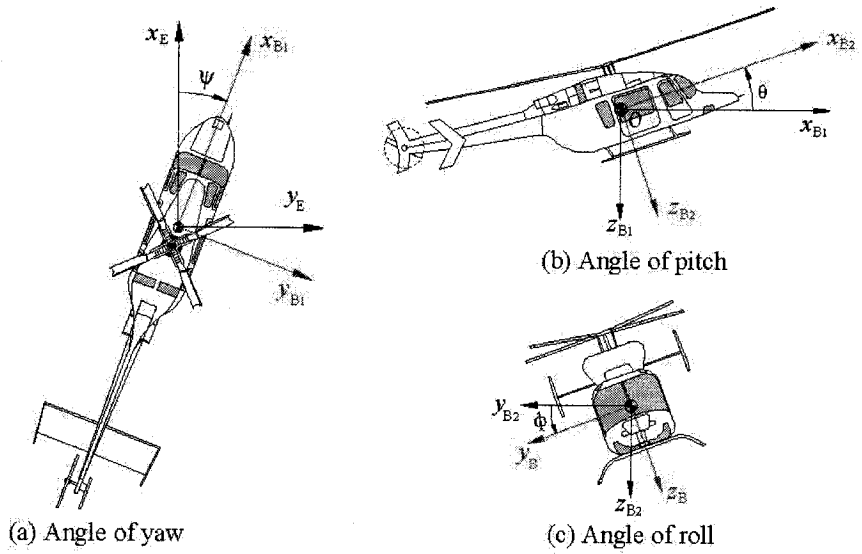


Figure 2-5 Euler angles of the helicopter

The first rotation involves the Euler angle ψ (Figure 2-5 (a)). The x_E, y_E, z_E axis system is rotated about the z_E-axis through a counter-clockwise angle ψ to give the new system $x_{B_1}, y_{B_1}, z_{B_1}$. It is clear from Figure 2-4 that this rotation mixes the coordinates along x_E and y_E, the coordinate along z_E remains unaffected. The rotation matrix to describe this operation is given by:

$$\left[{}^{E}_{B_1}R_{z_E}(\psi)\right] = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where $\left[{}^{E}_{B_1}R_{z_E}(\psi) \right]$ is the rotation matrix from frame $\{B_1\}$ to frame $\{E\}$, representing a rotation of an angle ψ about the z_E axis.

The second rotation involves the Euler angle θ (Figure 2-5 (b)). The $x_{B_1}, y_{B_1}, z_{B_1}$ axis system is rotated about the y_{B_1} axis through a counter-clockwise angle θ to generate the new coordinate system $x_{B_2}, y_{B_2}, z_{B_2}$. Analogous to the first Euler rotation, this mixes the coordinates along x_{B_1} and z_{B_1}, the coordinate along y_{B_1} remains unaffected. This operation also generates a line of nodes parallel to the direction y_{B_1}. The rotation matrix to describe this operation is given by:

$$\left[{}^{B_1}_{B_2}R_{y_{B_1}}(\theta) \right] = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

where $\left[{}^{B_1}_{B_2}R_{y_{B_1}}(\theta) \right]$ is the rotation matrix from frame $\{B_2\}$ to frame $\{B_1\}$, representing a rotation of an angle θ about the y_{B_1} axis.

The last rotation involves the Euler angle ϕ (Figure 2-5 (c)), the $x_{B_2}, y_{B_2}, z_{B_2}$ axis system is rotated about the z_{B_1} axis through a counter-clockwise angle ϕ to generate the final coordinate system x_B, y_B, z_B. Analogous to the first Euler rotation, this mixes the coordinates along x_{B_2} and y_{B_2} while the coordinate along z_{B_2} remains unaffected. The rotation matrix to describe this operation is given by:

$$\left[{}^{B_2}_{B}R_{x_{B_2}}(\phi)\right]=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sin\phi & \cos\phi\end{bmatrix}$$

where $\left[{}^{B_2}_{B}R_{x_{B_2}}(\phi)\right]$ is the rotation matrix from frame $\{B_2\}$ to frame $\{B\}$, representing a rotation of an angle ϕ about the x_{B_2} axis.

The combined effect of these three rotations is given by the following transformation matrix:

$$\left[{}^{E}_{B}R(\phi,\theta,\psi)\right]=\left[{}^{E}_{B_1}R_{z_E}(\psi)\right]\left[{}^{B_1}_{B_2}R_{y_{B_1}}(\theta)\right]\left[{}^{B_2}_{B}R_{x_{B_2}}(\phi)\right]$$

where $\left[{}^{E}_{B}R(\phi,\theta,\psi)\right]$ is the transformation matrix from frame {B} to frame {E} or equivalently

$$\left[{}^{E}_{B}R(\phi,\theta,\psi)\right]=\begin{bmatrix}\cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi-\sin\psi\cos\phi & \sin\psi\sin\phi+\cos\psi\sin\theta\cos\phi\\ \sin\psi\cos\theta & \cos\psi\cos\phi+\sin\psi\sin\theta\sin\phi & \sin\psi\sin\theta\cos\phi-\cos\psi\sin\phi\\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi\end{bmatrix} \quad (2.1)$$

Since $\left[{}^{E}_{B}R(\phi,\theta,\psi)\right]$ is the rotation matrix from frame $\{B\}$ to frame $\{E\}$, the linear velocity vector $\left[V_x,\ V_y,\ V_z\ \right]^T$ expressed in the inertial frame is related to the linear velocity vector $\left[u,\ v,\ w\ \right]^T$ expressed in the body-fixed frame in the form

$$\begin{bmatrix}V_x\\ V_y\\ V_z\end{bmatrix}=\left[{}^{E}_{B}R(\phi,\theta,\psi)\right]\begin{bmatrix}u\\ v\\ w\end{bmatrix}$$

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \qquad (2.2)$$

It is important to recall that the inverse of a rotation matrix is equal to the transpose matrix $\left[{}^{B}_{A}R\right] = \left[{}^{A}_{B}R\right]^{-1} = \left[{}^{A}_{B}R\right]^{T}$ since rotation matrices are orthonormal.

Equation (2.2) allows us to calculate the velocities V_x, V_y, V_z in earth system of coordinates when the velocities in the body system of coordinates u, v, w and the Euler angles ϕ, θ, ψ are known. The transformation from earth system coordinates to body system coordinates is $\left[{}^{B}_{E}R(\phi,\theta,\psi)\right]$ called the Direction Cosines Matrix [7], given by equation (2.3):

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \left[{}^{B}_{E}R(\phi,\theta,\psi)\right] \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \qquad (2.3)$$

where $\left[{}^{B}_{E}R(\phi,\theta,\psi)\right] = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \cos\theta\sin\phi \\ \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi & \cos\theta\cos\phi \end{bmatrix}$

In order to obtain the relationship between the angular velocity p, q, r and the rates $(\dot{\phi},\dot{\theta},\dot{\psi})$ of the angles of roll, pitch and yaw, we analyze figure 2-4. It can be seen that angular velocity $[p,\ q,\ r]^{T}$ can be expressed as:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + \left[{}^{B}_{B_2}R_{x_{B_2}}(\phi)\right] \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \left[{}^{B}_{B_2}R_{x_{B_2}}(\phi)\right] \left[{}^{B_2}_{B_1}R_{y_{B_1}}(\theta)\right] \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix}$$

Therefore, $[p,\ q,\ r]^T$ and $[\dot{\phi},\dot{\theta},\dot{\psi}]^T$ are related by

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.4}$$

The inverse relation can be written in the form

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.5}$$

Equation (2.4) shows that the angular velocities (p,q,r) of the helicopter can be determined by measuring the Euler angle (ϕ,θ,ψ) with respect to time. Equation (2.5) shows that the Euler angles (ϕ,θ,ψ) of the helicopter can be obtained by integrating the measured angular velocities (p,q,r).

2.3 Dynamic equations of motion

For a helicopter, the dynamic equations of motion are usually expressed in the body-fixed coordinate system. The reason for this is the fact that the aerodynamic forces have a natural expression in the body-fixed frame since they are caused by the body-air relative motion, and also the fact that the moment of inertia is constant in time if it is expressed in the body-fixed coordinate system.

The equations of motion for a helicopter can be derived from Newton's second law of motion. The time rates of change of linear and angular momentum are all taken with

respect to the inertial frame where we choose the earth coordinate system $\{E\}$. These laws can be expressed by two vector equations,

$$\sum \bar{F} = \frac{d}{dt}(m\bar{V})\,|_E \tag{2.6}$$

$$\sum \bar{M} = \frac{d\bar{H}}{dt}\,|_E \tag{2.7}$$

where $|_E$ indicate the time rate of change of vector with respect to the earth coordinate system $\{E\}$.

$\bar{F}$ is the external applied force vector;

$\bar{M}$ is the external applied moment vector about the center of gravity;

$\bar{V}$ is the true speed vector;

$\bar{H}$ is the angular momentum vector about the center of gravity;

Rigorously, equation (2.7) can be applied only to a constant-mass system.

The external forces and moments consist of initial forces and moments, and changes of which may be caused by disturbances from initial conditions. Thus,

$$\sum \bar{F} = \sum \bar{F}_i + \sum \Delta\bar{F} \tag{2.8}$$

$$\sum \bar{M} = \sum \bar{M}_i + \sum \Delta\bar{M} \tag{2.9}$$

where $\sum \bar{F}_i$, $\sum \bar{M}_i$ are the summations of the initial force and moment vectors. Equations (2.6) to (2.9) in the earth coordinate system $\{E\}$ need to be transformed to the rotating helicopter body axis system. From the equations of Coriolis [6], we can relate the derivative of a vector in two different frames as follows:

$$\frac{d\bar{V}}{dt}\Big|_E = \frac{d\bar{V}}{dt}\Big|_B + (\bar{\omega}\times\bar{V})_B \tag{2.10}$$

$$\frac{d\bar{H}}{dt}\Big|_E = \frac{d\bar{H}}{dt}\Big|_B + (\bar{\omega}\times\bar{H})_B \tag{2.11}$$

By replacing equation (2.10), (2.11) into (2.6), (2.7), we obtain:

$$\sum \bar{F} = m\left[\frac{d\bar{V}}{dt}\Big|_B + (\bar{\omega}\times\bar{V})_B\right] \tag{2.12}$$

$$\sum \bar{M} = \frac{d\bar{H}}{dt}\Big|_B + (\bar{\omega}\times\bar{H})_B \tag{2.13}$$

$\bar{\omega}$ is the angular velocity vector of the body-axis system with respect to the earth coordinate system. Its components in the body-axis system are (p, q, r). The components of $\bar{V}$ in the body-axis system are (u, v, w).

$$(\bar{\omega}\times\bar{V})_B = \begin{vmatrix} \bar{i} & \bar{j} & \bar{k} \\ p & q & r \\ u & v & w \end{vmatrix} = (wq - vr)\ \bar{i} + (ur - wp)\ \bar{j} + (vp - uq)\ \bar{k} \tag{2.14}$$

where $\bar{i}, \bar{j}, \bar{k}$ are unit vectors along the helicopter's body axes respectively. The angular momentum with respect to the body-axis system is as:

$$\begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix} = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.15}$$

where I_x, I_y, I_z, I_{xy}, I_{xz}, I_{yz} are the moments of inertia of the helicopter.

Equation (2.12) and (2.13) can be written in the following scalar form:

Force equation:

$$\left.\begin{aligned} F_x &= m(\dot{u} + wq - vr) \\ F_y &= m(\dot{v} + ur - wp) \\ F_z &= m(\dot{w} + vp - uq) \end{aligned}\right\} \tag{2.16}$$

where F_x, F_y, F_z are the components of the external applied forces.

Moment equations

$$\left.\begin{aligned} L &= \dot{p}I_x - \dot{q}I_{xy} - \dot{r}I_{xy} + qr(I_z - I_y) + (r^2 - q^2)I_{yz} - pqI_{xz} + rpI_{xy} \\ M &= -\dot{p}I_{xy} + \dot{q}I_y - \dot{r}I_{yz} + pr(I_x - I_z) + (p^2 - r^2)I_{xz} - qrI_{xy} + pqI_{yz} \\ N &= -\dot{p}I_{xz} - \dot{q}I_{yz} + \dot{r}I_z + pq(I_y - I_x) + (q^2 - p^2)I_{xy} - rpI_{yz} + qrI_{xz} \end{aligned}\right\} \tag{2.17}$$

where L, M and N are the components of the external applied moments.

The helicopter has an approximately symmetrical mass distribution relative to the $x_B z_B$-body plane, hence, I_{xy} and I_{yz} are far smaller than the four moments of inertias (I_x, I_y, I_z and I_{xz}), thus they can be omitted. The general inertial-moment equation (2.17) can be written:

$$\left.\begin{aligned} L &= I_x\dot{p} + (I_z - I_y)qr - I_{zx}(\dot{r} + pq) \\ M &= I_y\dot{q} + (I_x - I_z)pr + I_{zx}(p^2 - r^2) \\ N &= I_z\dot{r} + (I_y - I_x)pq + I_{zx}(qr - \dot{p}) \end{aligned}\right\} \tag{2.18}$$

The components of the gravity force along the helicopter axes are

x_B-axis : $-mg\sin\theta$

y_B-axis: $mg\cos\theta\sin\phi$

z_B-axis: $mg\cos\theta\cos\phi$

Considering F_x, F_y and F_z as functions of the gravity forces and aerodynamic forces X, Y, and Z along the body coordinate axes of x_B, y_B, z_B respectively, and substituting the forces into equation (2.16), we have the aerodynamic forces:

$$\left.\begin{aligned} X &= m\dot{u} + m(wq - vr) + mg\sin\theta \\ Y &= m\dot{v} + m(ur - wp) - mg\cos\theta\sin\phi \\ Z &= m\dot{w} + m(-uq + vp) - mg\cos\theta\cos\phi \end{aligned}\right\} \quad (2.19)$$

Equations (2.18) describe the rotational dynamics of a helicopter as a rigid body and equations (2.19) represent the translational dynamics. Both equations are nonlinear due to the appearance of products of angular rates and the gravitational and rotational related terms in the force equations (2.19). The set of equations (2.18) and (2.19) is called the ***helicopter dynamics***; it is a 6 DoF (six degrees of freedom) model.

2.4 Flight control vector

For a helicopter, the control vector has four components: ***longitudinal cyclic***, ***main rotor collective*** (***collective***), ***lateral cyclic,*** and ***tail rotor collective*** (***pedals***). It is expressed as:

$$\bar{u}(t) = \left[\delta_{lon}, \delta_{coll}, \delta_{lat}, \delta_{ped}\right] \quad (2.20)$$

The control vector can be changed by the pilot's operation. When the control vector changes, it generates off-trim forces and moments.

The longitudinal cyclic controls the longitudinal cyclic pitch of the main rotor blades which creates a longitudinal tilt of the thrust vector, and causes a pitch and a translational motion such as climbing, forward flight, and descent flight in longitudinal plane.

The collective controls the collective pitch of the main rotor blades which increases or decreases the thrust vector, which in turn creates a translation motion along the z_B axis. Due to the anti-coupling effect of the main rotor, the collective controls increase or decrease the yaw moment which creates a yaw motion along the z_B axis.

The lateral cyclic controls the lateral cyclic pitch of the main rotor blades which creates a lateral tilt of the thrust vector, and causes a roll and a translational motion in the lateral plane.

The pedals control the collective pitch of the tail rotor blades which creates a yaw moment and a yaw motion along the z_B axis.

2.5 Linearization of helicopter dynamics

The essence of linearization is the assumption that the helicopter motion can be considered as a small perturbation about a trim or equilibrium condition. This fundamental assumption of linearization makes external forces X, Y and Z and moments L, M and N representable as analytic functions of the perturbed motion variables and their derivative.

For the 6 DoF helicopter dynamics, we re-write equations (2.18), (2.19) and putting them together with (2.5) and (2.20), we have a set of equations for the helicopter dynamics model:

$$\left.\begin{aligned} \dot{u} &= \tilde{X} - (wq - vr) - g\sin\theta \\ \dot{v} &= \tilde{Y} - (ur - wp) + g\cos\theta\sin\phi \\ \dot{w} &= \tilde{Z} - (vp - uq) + g\cos\theta\cos\phi \end{aligned}\right\} \tag{2.21}$$

where $\tilde{X} = \frac{X}{m}$, $\tilde{Y} = \frac{Y}{m}$, $\tilde{Z} = \frac{Z}{m}$ are normalized forces. For convenience, we use X, Y, Z instead of $\tilde{X}$, $\tilde{Y}$, $\tilde{Z}$ in what follows.

$$\left.\begin{aligned} \dot{p} &= \frac{1}{I_x}L + \frac{(I_y - I_z)}{I_x}qr + \frac{I_{zx}}{I_x}(\dot{r} + pq) \\ \dot{q} &= \frac{1}{I_y}M + \frac{(I_z - I_x)}{I_y}pr + \frac{I_{zx}}{I_y}(r^2 - p^2) \\ \dot{r} &= \frac{1}{I_z}N + \frac{(I_x - I_y)}{I_z}pq + \frac{I_{zx}}{I_z}(\dot{p} - qr) \end{aligned}\right\} \tag{2.22}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.23}$$

$$\overline{u}(t) = \left[\delta_{lon}, \delta_{coll}, \delta_{lat}, \delta_{ped}\right] \tag{2.24}$$

We commonly write the nonlinear differential equations in the first-order vector form,

$$\frac{d\bar{x}}{dt} = \bar{F}(\bar{x}, \bar{u}, t) \tag{2.25}$$

with initial conditions $\bar{x}(0) = \bar{x}_0$, where $\bar{x}(t)$ is the vector of the state variable and $\bar{u}(t)$ is the control vector. Equation (2.25) includes three fundamental problems of flight dynamics: trim, stability and response.

The solutions of the trim, stability and response of the flight dynamics can be written as:

Trim: $$\bar{F}(\bar{x}_e, \bar{u}_e) = 0 \tag{2.26}$$

Stability: $$\det\left[\lambda I - \left(\frac{\partial \bar{F}}{\partial \bar{x}}\right)_{\bar{x}_e}\right] = 0 \tag{2.27}$$

Response: $$\bar{x}(t) = \bar{x}(0) + \int_0 \bar{F}(\bar{x}(\tau), \bar{u}(\tau), \tau)\ d\tau \tag{2.28}$$

The trim solution of equation (2.26) is represented by the zero of a nonlinear algebraic function, where the controls $\bar{u}_e$ required to hold a defined state $\bar{x}_e$ (subscript e denotes equilibrium) are computed. A trimmed flight condition is an unaccelerated flight and the resultant of the applied forces and moments is zero. In a trimmed manoeuvre, the helicopter will accelerate under the action of non-zero resultant aerodynamic and gravitational forces and moments, but these will then be balanced by the effects of centrifugal and gyroscopic inertial forces and moments.

The stability solution of equation (2.27) is found by linearizing the equations about a particular trim condition and computing the eigenvalues of the helicopter system matrix. The stability is determined by the signs of the real parts of the eigenvalues λ : if the signs of the real parts of all the eigenvalues are negative, the helicopter is stable, otherwise the helicopter is unstable. Stability is concerned with the tendency of an aircraft to return to its trim conditions following a disturbance. The static stability is defined as the tendency of an aircraft to return to its trim condition following changes in forces or moments. The dynamic stability deals with the oscillation of an aircraft about its trim position following a disturbance from trim. The stability characteristics of a system can be categorized by the type of time history it has following a displacement [8].

The response solution of equation (2.28) depends on the initial conditions of the motion state vector and the time variation of the vector function $\overline{F}(\overline{x},\overline{u},t)$, that includes the aerodynamic loads, gravitational forces and inertial forces and moments. The trajectory can be computed using different numerical integration schemes which time march through a simulation, achieving an approximate balance of the component accelerations with the applied forces and moments at every time step. This is an efficient process for solving equation (2.25), but numerical integration offers little insight into the physics of the helicopter behavior. We need analytic solutions to deliver a deeper understanding between cause and effect; but the scope for deriving analytic solutions of general nonlinear differential equations as in equation (2.25) is extremely limited; functional forms can only be found in special cases, and even then, the range of validity is likely to be very small. Therefore, we linearize equation (2.25) and study linear

approximations to the general nonlinear motion in order to analyze the complex dynamic aircraft motions.

For an airplane—which has lateral symmetry—the six equation of equilibrium can be conveniently dealt with in two groups; the longitudinal equations consisting of X, Z, and M; and the lateral directional equations consisting of Y, L, and N. A helicopter is not quite as symmetrical as an airplane, and there are stronger cross-coupling effects such as lateral and longitudinal coupling. Thus, from a rigorous point of view, the trim equations should be determined from a simultaneous solution of all 6 DoF helicopter dynamics; but from a practical point of view, it is necessary to simplify the motion equations of the helicopter [9]. For this purpose, we make use of common assumptions as follows [1]:

(1) The rotor speed remains constant.

(2) The motion is a sequence of steady conditions, i.e. the accelerations of the helicopter are small enough to have a negligible effect on the rotor response.

Based on these assumptions, the helicopter is considered to be in equilibrium (trim conditions) before a disturbance is introduced; the disturbances in general arise from either control surface deflection or atmospheric turbulence. The assumptions are made that the aerodynamic forces and moments can be expressed as a multi-dimensional analytic function of the motion of the aircraft about the trim condition.

Equations (2.21) to (2.23) are non-linear. In the dynamic analysis, we apply small perturbation theory to linearize the helicopter dynamics.

We suppose that the helicopter is in a trimmed flight condition, and then is disturbed by a small perturbation which comes from wind or operation of pilot, during disturbed

motion the helicopter behavior can be described as a perturbation from the trim. In 6 DoF form, the motion state is $\bar{x}(t)=[u,w,q,\theta,v,p,\phi,r,\psi]^T$. Its components and the forces and moments can be expressed during disturbances as follows:

$$\left.\begin{array}{llll} X=X_e+\Delta X, & Y=Y_e+\Delta Y, & Z=Z_e+\Delta Z, & L=L_e+\Delta L, \\ M=M_e+\Delta M, & N=N_e+\Delta N, & u=u_e+\Delta u, & v=v_e+\Delta v, \\ w=w_e+\Delta w & p=p_e+\Delta p, & q=q_e+\Delta q, & r=r_e+\Delta r, \\ \phi=\phi_e+\Delta\phi, & \theta=\theta_e+\Delta\theta, & \psi=\psi_e+\Delta\psi, & \end{array}\right\} \quad (2.29)$$

where u_e, v_e, etc. are the initial values and Δu, Δv, etc. are the changes in these values resulting from some disturbance. The aerodynamic forces are function of $\bar{x}(t)=[u,w,q,\theta,v,p,\phi,r,\psi]^T$ and the control vector $\bar{u}(t)=[\delta_{long},\ \delta_{lat},\ \delta_{ped},\ \delta_{col}]^T$. We use Taylor's theorem and neglect second and higher order terms, and thus obtain the following expression of aerodynamic force ΔY

$$\begin{aligned} Y=Y_e&+\frac{\partial Y}{\partial u}\Delta u+\frac{\partial Y}{\partial w}\Delta w+\frac{\partial Y}{\partial q}\Delta q+\frac{\partial Y}{\partial \theta}\Delta\theta+\frac{\partial Y}{\partial v}\Delta v+\frac{\partial Y}{\partial p}\Delta p+\frac{\partial Y}{\partial \phi}\Delta\phi+\frac{\partial Y}{\partial r}\Delta r+\frac{\partial Y}{\partial \psi}\Delta\psi \\ &+\frac{\partial Y}{\partial \delta_{long}}\delta_{long}+\frac{\partial Y}{\partial \delta_{lat}}\delta_{lat}+\frac{\partial Y}{\partial \delta_{ped}}\delta_{ped}+\frac{\partial Y}{\partial \delta_{col}}\delta_{col} \end{aligned} \quad (2.30)$$

Dropping Δ on the right-hand side, i.e. the disturbed variables Δu, Δv,... are replaced by u, v, ...and introducing the standard stability and control derivative notation

$$Y_u=\frac{\partial Y}{\partial u},\quad Y_v=\frac{\partial Y}{\partial v},\quad Y_w=\frac{\partial Y}{\partial w}\ \ \ldots,\quad Y_{\delta_{col}}=\frac{\partial Y}{\partial \delta_{col}}\ \ \ldots$$

Equation (2.30) becomes:

$$\Delta Y = Y_u u + Y_w w + Y_q q + Y_\theta \theta + Y_v v + Y_p p + Y_\phi \phi + Y_r r + Y_\psi \psi$$
$$+ Y_{long}\delta_{long} + Y_{coll}\delta_{coll} + Y_{lat}\delta_{lat} + Y_{ped}\delta_{ped} \quad (2.31)$$

Applying the same analysis to the other forces and moments acting on a helicopter, we obtain the following equation about stability derivatives and control derivatives,

$$\begin{bmatrix}\Delta X\\ \Delta Y\\ \Delta Z\\ \Delta L\\ \Delta M\\ \Delta N\end{bmatrix} = \begin{bmatrix} X_u & X_w & X_q & X_\theta & X_v & X_p & X_\phi & X_r & X_\psi \\ Y_u & Y_w & Y_q & Y_\theta & Y_v & Y_p & Y_\phi & Y_r & Y_\psi \\ Z_u & Z_w & Z_q & Z_\theta & Z_v & Z_p & Z_\phi & Z_r & Z_\psi \\ L_u & L_w & L_q & L_\theta & L_v & L_p & L_\phi & L_r & L_\psi \\ M_u & M_w & M_q & M_\theta & M_v & M_p & M_\phi & M_r & M_\psi \\ N_u & N_w & N_q & N_\theta & N_v & N_p & N_\phi & N_r & N_\psi \end{bmatrix} \bar{x}(t) + \begin{bmatrix} X_{\delta_{long}} & X_{\delta_{coll}} & X_{\delta_{lat}} & X_{\delta_{ped}} \\ Y_{\delta_{long}} & Y_{\delta_{coll}} & Y_{\delta_{lat}} & Y_{\delta_{ped}} \\ Z_{\delta_{long}} & Z_{\delta_{coll}} & Z_{\delta_{lat}} & Z_{\delta_{ped}} \\ L_{\delta_{long}} & L_{\delta_{coll}} & L_{\delta_{lat}} & L_{\delta_{ped}} \\ M_{\delta_{long}} & M_{\delta_{coll}} & M_{\delta_{lat}} & M_{\delta_{ped}} \\ N_{\delta_{long}} & N_{\delta_{coll}} & N_{\delta_{lat}} & N_{\delta_{ped}} \end{bmatrix} \bar{u}(t) \quad (2.32)$$

If the force and moment functions (i.e. the aerodynamic loads) and all their derivatives are known at any point (the trim condition), then the behavior of that function anywhere in its analytic range can be estimated from an expansion of the function in a series about the known point.

Substituting equations (2.29) and (2.32) into (2.21) to (2.23) yields the linearized equations of motion for the full 6 DoF. Considering atmospheric and other disturbances expressed by $\bar{f}(t)$, we obtain the equation (2.33) describing the perturbed motion about a general trim condition as follows:

$$\dot{\bar{x}} = A\bar{x} + B\bar{u}(t) + \bar{f}(t) \quad (2.33)$$

The matrix A is given by the equation (2.33b); it can be divided into four submatrices and expressed as:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{2.33a}$$

$$A = \left[\begin{array}{cccc:cccc} X_u & X_w-q_e & X_q-w_e & -g\cos\theta_e & X_v+r_e & X_p & 0 & X_r+v_e \\ Z_u+q_e & Z_w & Z_q+u_e & -g\cos\phi_e\sin\theta_e & Z_v-p_e & Z_p-v_e & -g\sin\phi_e\cos\theta_e & Z_r \\ M_u & M_w & M_q & 0 & M_v & \begin{matrix}M_p-2p_eI_{xz}/I_y \\ -r_e(I_x-I_z)/I_y\end{matrix} & 0 & \begin{matrix}M_r+2r_eI_{xz}/I_y \\ -p_e(I_x-I_z)/I_y\end{matrix} \\ 0 & 0 & \cos\phi_e & 0 & 0 & 0 & -\dot{\psi}_e\cos\theta_e & -\sin\phi_e \\ \hdashline Y_u-r_e & Y_w+p_e & Y_q & -g\sin\phi_e\sin\theta_e & Y_v & Y_p+w_e & g\cos\phi_e\cos\theta_e & Y_r-u_e \\ L'_u & L'_w & \begin{matrix}L'_q+k_1p_e \\ -k_2r_e\end{matrix} & 0 & L'_v & L'_p+k_1q_e & 0 & L'_r-k_2q_e \\ 0 & 0 & \sin\phi_e\tan\theta_e & \dot{\psi}_e/\cos\theta_e & 0 & 1 & \dot{\theta}_e\tan\theta_e & \cos\phi_e\tan\theta_e \\ N'_u & N'_w & \begin{matrix}N'_q-k_1r_e \\ -k_3p_e\end{matrix} & 0 & N'_v & N'_p-k_3q_e & 0 & N'_r-k_1q_e \end{array}\right] \tag{2.33b}$$

$$B = \begin{bmatrix} X_{\delta_{long}} & X_{\delta_{coll}} & X_{\delta_{lat}} & X_{\delta_{ped}} \\ Z_{\delta_{long}} & Z_{\delta_{coll}} & Z_{\delta_{lat}} & Z_{\delta_{ped}} \\ M_{\delta_{long}} & M_{\delta_{coll}} & M_{\delta_{lat}} & M_{\delta_{ped}} \\ 0 & 0 & 0 & 0 \\ Y_{\delta_{long}} & Y_{\delta_{coll}} & Y_{\delta_{lat}} & Y_{\delta_{ped}} \\ L_{\delta_{long}} & L_{\delta_{coll}} & L_{\delta_{lat}} & L_{\delta_{ped}} \\ 0 & 0 & 0 & 0 \\ N_{\delta_{long}} & N_{\delta_{coll}} & N_{\delta_{lat}} & N_{\delta_{ped}} \end{bmatrix} \tag{2.33c}$$

$$k_1 \equiv \frac{I_{xz}(I_x+I_z-I_y)}{I_xI_z-I_{xz}^2}, \qquad k_2 \equiv \frac{I_z(I_z-I_y)+I_{xz}^2}{I_xI_z-I_{xz}^2}, \qquad k_3 \equiv \frac{I_x(I_y-I_x)-I_{xz}^2}{I_xI_z-I_{xz}^2},$$

$$L'_- \equiv \frac{I_zL_- + I_zN_-}{I_xI_z-I_{xz}^2}, \qquad N'_- \equiv \frac{I_{xz}L_- + I_xN_-}{I_xI_z-I_{xz}^2}.$$

The index "_" could be any of u, w, q, v, p, r. There are 36 stability derivatives in matrix A, and 24 control derivatives in matrix B. The derivatives are described in Table 2.1. The most commonly used stability and control derivatives in Table 2.1 have acquired descriptors based on their effect on the stability and control characteristics as presented in Table 2.2 [see reference 2].

Table 2.1 The stability and control derivatives description

Derivative	description
X_u, Y_u, Z_u	Forward, side and vertical forces due to forward velocity
X_v, Y_v, Z_v	Forward, side and vertical forces due to lateral velocity
X_w, Y_w, Z_w	Forward, side and vertical forces due to vertical velocity
L_u, M_u, N_u	Rolling, pitching and yawing moments due to forward velocity
L_v, M_v, N_v	Rolling, pitching and yawing moments due to lateral velocity
L_w, M_w, N_w	Rolling, pitching and yawing moments due to vertical velocity
X_p, Y_p, Z_p	Forward, side and vertical forces due to roll rate
X_q, Y_q, Z_q	Forward, side and vertical forces due to pitch rate
X_r, Y_r, Z_r	Forward, side and vertical forces due to yaw rate
L_p, M_p, N_p	Rolling, pitching and yawing moments due to roll rate
L_q, M_q, N_q	Rolling, pitching and yawing moments due to pitch rate
L_r, M_r, N_r	Rolling, pitching and yawing moments due to yaw rate
$X_{\delta_{long}}, Y_{\delta_{long}}, Z_{\delta_{long}}$	Forward, side and vertical forces due to longitudinal cyclic pitch
$X_{\delta_{coll}}, Y_{\delta_{coll}}, Z_{\delta_{coll}}$	Forward, side and vertical forces due to collective pitch
$X_{\delta_{lat}}, Y_{\delta_{lat}}, Z_{\delta_{lat}}$	Forward, side and vertical forces due to lateral cyclic pitch
$X_{\delta_{ped}}, Y_{\delta_{ped}}, Z_{\delta_{ped}}$	Forward, side and vertical forces due to tail rotor pitch
$L_{\delta_{long}}, M_{\delta_{long}}, N_{\delta_{long}}$	Rolling, pitching and yawing moments due to longitudinal cyclic pitch
$L_{\delta_{coll}}, M_{\delta_{coll}}, N_{\delta_{coll}}$	Rolling, pitching and yawing moments due to collective pitch
$L_{\delta_{lat}}, M_{\delta_{lat}}, N_{\delta_{lat}}$	Rolling, pitching and yawing moments due to lateral cyclic pitch
$L_{\delta_{ped}}, M_{\delta_{ped}}, N_{\delta_{ped}}$	Rolling, pitching and yawing moments due to tail rotor pitch

Table 2.2 The most commonly used stability and control derivatives

Derivative	Descriptor	Derivative	Descriptor
X_u	Drag damping	L_p	Roll damping
Y_v	Side force	M_q	Pitch damping
Z_w	Heave damping	N_r	Yaw damping
L_v	Lateral static stability	L_{lat}	Roll control power
M_u	Speed stability	M_{lon}	Pitch control power
M_w	Angle of attack stability	N_{ped}	Yaw control power
N_v	Directional static stability	Z_{col}	Heave control power
L_{ped}	Tail rotor roll	Y_{ped}	Tail rotor drift
M_{col}	Pitch change with power	N_{col}	Torque reaction

In equation (2.33), B is the control matrix. Submatrix A_{11} gives the longitudinal equations of motion, while submatrix A_{22} gives the lateral-direction equations of motion. The other two submatrices A_{12}, A_{21} represent the coupling between the primary submatrices. In matrix A, the heading angle ψ has been omitted because the direction of flight in the horizontal plane has no effect on the aerodynamics of forces and moments. In addition to the linearized aerodynamic forces and moments, A also contains perturbational inertial, gravitational and kinematic effects linearized about the trim condition defined by $u_e, v_e, w_e, p_e, q_e, r_e, \phi_e, \theta_e$. The heading angle velocity can be derived from equation (2.23) by use of small perturbation theory as follows:

$$\dot{\psi} = \frac{\sin\phi_e}{\cos\theta_e} q + (\dot{\psi}_e \tan\theta_e)\theta + \frac{\dot{\theta}_e}{\cos\theta_e}\phi + \frac{\cos\phi_e}{\cos\theta_e} r \tag{2.36}$$

Now, we obtain the helicopter linearized equations of motion. These can be written as a state space system which is used to define the helicopter's motion as follows:

$$\left.\begin{aligned}\dot{\bar{x}}(t) &= A\bar{x}(t) + B\bar{u}(t) + \bar{f}(t)\\ \bar{y}(t) &= C\bar{x}(t) + D\bar{u}(t)\end{aligned}\right\} \tag{2.37}$$

where C and D matrices represent the linear relationships between the output vector $\bar{y}$ and the state vector $\bar{x}$. A and B are called stability matrix and control matrix, respectively.

Equation (2.37) can be simplified into several distinct simulation models for various specific helicopter flight conditions such as take off, hover, level flight, climbing flight, descent flight, autorotation, and landing. The Composition of the models is the helicopter global simulation model.

When matrices A and B contain the unknown parameters representing stability and control derivatives, equation (2.37) is called the helicopter parameter identification model. Parameter identification of the helicopter simulation model is to find the unknown values from the flight test data by use of the Maximum Likelihood Estimation (MLE). Usually, a set of solutions are obtained rather than a unique solution due to the strong coupling of parameters. The selected solution is submitted into equation (2.37); at this moment, equation (2.37) becomes the simulation model which will be validated and is called the helicopter validation model.

Helicopter simulation model validation is to evaluate and approve whether the selected solution by MLE is reasonable from the flight test data. If it is reasonable, the simulated results from the simulation model should match with the flight data for the same flight conditions. Otherwise, another solution should be selected. Considering the

disturbances, the simulated results should be within the tolerances described in the Advisory Circular AC 120-63 [15]. The validation process will be presented in Chapter 4.

CHAPTER 3

FLIGHT TESTS AND CORRECTIONS OF THE RAW DATA

Flight testing is the process of gathering information or data which will accurately describe the capabilities of a particular type of aircraft, and which can be used to accurately predict and optimize the use of all aircrafts of that same type in future missions. In this chapter, we introduce flight test, instrumentations, measurement of flight data and the correction of the raw flight data. We derive the true air speed (TAS) at the *CG*, the transformation expressions of accelerations and airspeed from the Instrumentation Center (IC) to the Center of Gravity (CG).

3.1 Introduction of flight testing

The purpose of flight testing is to gather data and build a database about the flight characteristics of an aircraft and its subsystems for subsequent analysis on the ground. This database includes the parameters recorded during a series of flight tests of the helicopter in different flight conditions such as loading and CG positions, rotor speeds, flight mission and manoeuvres etc.; it also covers the entire flight envelope. The database contains two parts; the first part is a set of 2311 control input manoeuver flight tests which is used to identify stability and control derivatives, the second part is a set of flight tests recorded in the same conditions as the first part, which is used for the simulation model validation.

In order to estimate the stability and control derivatives from flight test data, it is essential to excite the helicopter motion by applying appropriate control inputs to the longitudinal, lateral, collective cyclic stick and the pedal successively. Generally, there are two forms of control inputs sequences which are used in helicopter flight tests, they are the 3211 input signal and 2311 input signal [10]. Figure 3-1 shows the 3211 and 2311 input signals. For Bell 427 parameter estimation program, the 2311 control input sequence is used because (1) it is ability to excite all the natural modes. (2) it is short time duration and easy to execute and repeat, (3) it is sufficiently high frequency content, provided by the alternating input strokes, in order to improve control derivative estimation.

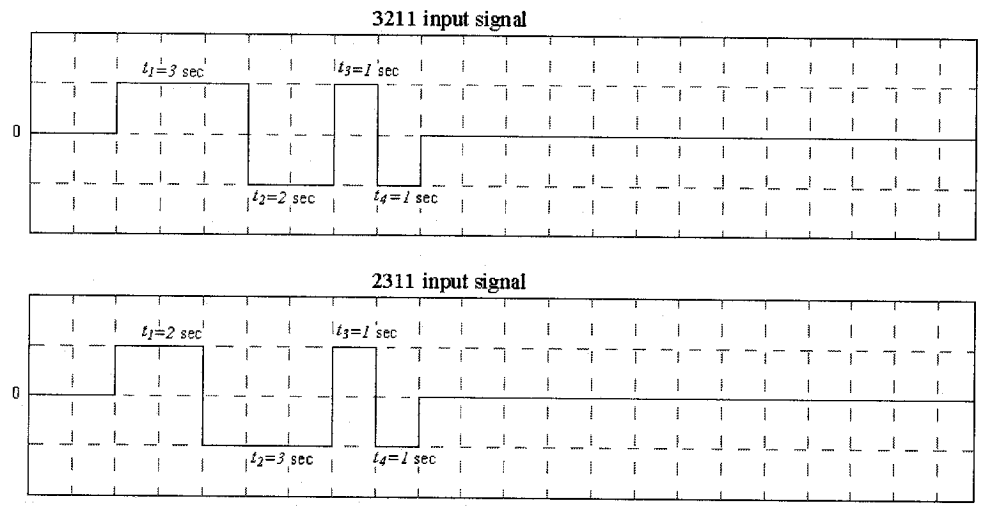


Figure 3-1 3211 and 2311 input signals.

The recorded data is preprocessed, corrected and organized in standardized formats so that all partners involved in the project can easily use it. These restructured data are used in parameter estimation and in the simulation model validation.

The test data gathering process starts with sensors or transducers which have been mounted in the helicopter. Different kinds of transducers are used to measure control positions, pressures, temperatures, etc.

The electrical signal from each transducer is sent through special instrumentation wiring to a central location in the helicopter where all the electrical signals are formatted, organized and recorded, then transferred to a telemetry transmitter. The telemetry transmitter transmits the data from the aircraft to a ground station. The ground station receives and records all of the data, and converts portions of the data stream into electrical signals that can be displayed on indicators in the ground control room. In this way engineers on the ground can monitor flight activities and can assist the pilot in the safe conduct of the flight.

The aim of obtaining flight test data is to extract stability and control derivatives of the helicopter by using parameter estimation methods. Since high accuracy of flight test data is a prerequisite for system identification, the helicopter should be instrumented with high quality sensors mainly to provide data of linear accelerations, rates, attitudes, speed components and control positions. The Bell 427 flight test data for system identification and simulation purpose are obtained from the following devices.

(1) Nose boom

The nose boom is installed in the front of the helicopter fuselage; this device is used to measure airspeed, pressure, altitude, angle of attack α and angle of sideslip β. To avoid the main rotor's wake interaction, the sensors are located where the measurements are not influenced by the shape and the wake of the helicopter. Figure 3-2 shows the

location of a nose boom of a helicopter. The Pitot tubes are used to measure speed and altitude and are located in the front of the nose boom. They include two pressure measurements, total pressure (or pitot pressure) and static pressure. The static pressure measurement can be related directly to the pressure altitude at which the helicopter is flying. The difference between the total and static pressure can be related to the speed of the helicopter through the air [11]. Behind the pitot tubes on the nose boom are two vanes (very much like miniature weather vanes) that pivot freely on posts extending vertically and horizontally from the nose boom. The vanes align themselves with the free stream airflow. A transducer measures the position of the two vanes relative to the fuselage centerline. The horizontal vane (alpha vane) and vertical vane (beta vane) are used to record the angles α and β, respectively, and the True Air Speed (TAS) orientations with respect to the body-fixed frame. Both are key measurements for determining the helicopter stability.

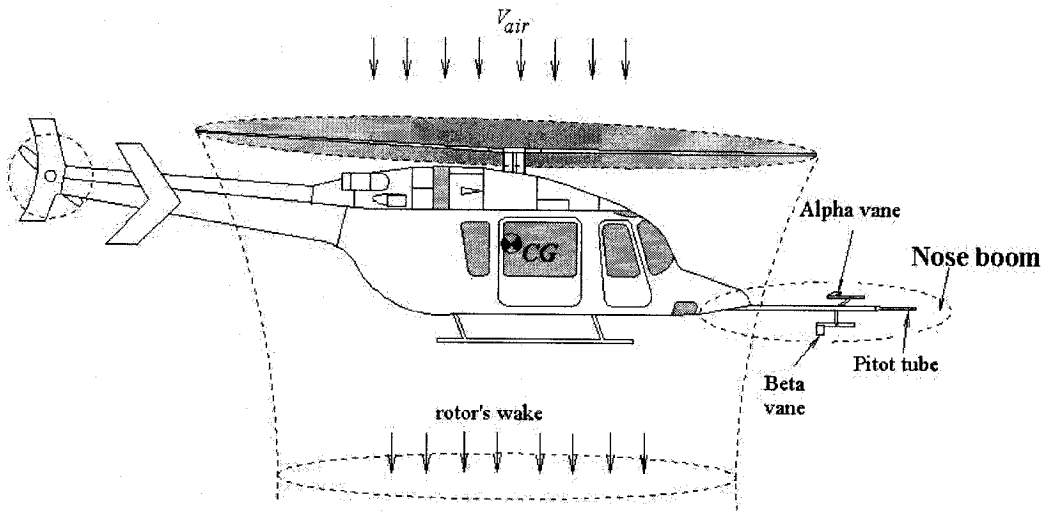


Figure 3-2 Location of a nose boom of the helicopter

(2) Gyros and Accelerometers [12]

A rate gyroscope (Gyro) measures the rates of rotation (p,q,r) about the three body-fixed axes x_B, y_B, z_B. Accelerometers measure the linear accelerations along the three axes on which they are located. A gyrocompass measures yaw attitude ψ, while a vertical gyroscope records the pitch attitude ϕ and roll attitude θ. The accelerometers and three gyros are usually very carefully aligned and mounted near the aircraft's center of gravity, often on the same mounting platform which is defined as the Instrumentation Center (*IC*).

(3) Other required instrumentation in the helicopter.

Potentiometers are installed to measure the control input $(\delta_{long}, \delta_{lat}, \delta_{ped}, \delta_{col})$, and an Outside Air Temperature (OAT) to record the temperature.

Table 3.1 shows a portion of the parameters used for identification and validation. Some parameters are directly recorded by the instrumentations, while some are derived from the recorded parameters. All parameters are used for further analysis. Locations of the sensors with respect to the center of gravity in the body-fixed frame are also recorded in order to correct the measured values from the locations of sensors to the Center of Gravity (*CG*).

3.2 Correction of the raw data

The reasons for raw data corrections are the following:

(1) Instruments offsetting from their theoretical values cause errors between the true data and the recorded data;

Table 3.1 Recorded and derived parameters during flight testing

Parameters	Units	Positive signs	Notes
Pressure altitude	ft	Positive	Recorded
Altitude	ft	Under sea level	Recorded
Outside air temperature	°C	Over freezing point	Recorded
Roll angle (ϕ)	deg	A/C turns clockwise about roll axis as seen from rear	Recorded
Pitch angle (θ)	deg	A/C nose up	Recorded
Yaw angle (ψ)	deg	A/C turns clockwise about yaw axis as seen from above	Recorded
Angle of Attack (α)	deg	A/C nose moves up	Derive from nose boom
Side slip angle (β)	deg	A/C nose moves to the left	Derive from nose boom
Roll rate (p)	deg/s	A/C turns clockwise about roll axis as seen from rear	Recorded
Pitch rate (q)	deg/s	A/C nose up	Recorded
Yaw rate (r)	deg/s	A/C turns clockwise about yaw axis as seen from above	Recorded
Longitudinal acceleration ($a_{x_{CG}}$)	m/s^2	Forward	Derive from the *IC*
Vertical acceleration ($a_{y_{CG}}$)	m/s^2	To the right	Derive from the *IC*
Lateral acceleration ($a_{z_{CG}}$)	m/s^2	Down	Derive from the *IC*
Longitudinal cyclic stick position(δ_{long})	%	Cyclic stick moves forward	Recorded
Lateral cyclic stick position(δ_{lat})	%	Cyclic stick moves to the right	Recorded
Pedal position (δ_{ped})	%	Right pedal moves forward	Recorded
Collective position (δ_{col})	%	Blade angle increases	Recorded
Indicated airs peed (IAS)	knots	Positive	Recorded
True air speed (TAS) of the *CG*	knots	Positive	Derive from IAS
Mach number (Mach)	-	Positive	Derive from TAS
Longitudinal speed (u_{CG})	ft/s	Forward	Derive from nose boom
Vertical speed (v_{CG})	ft/s	To the right	Derive from nose boom
Lateral speed (w_{CG})	ft/s	Down	Derive from nose boom

(2) The time delays of instrumentations cause the time delays of measured

parameters;

(3) Some of the raw data need to be presented in the appropriate units for U.S. operations, such as fuel in lb, speed in knots, altitudes in ft, etc.

(4) Some of the raw data need to be transferred from the *IC* to the *CG* because the *IC* and some sensors such as the alpha vane, etc. are located at different points with respect to the *CG*.

3.2.1 Transformation of accelerations from IC to CG

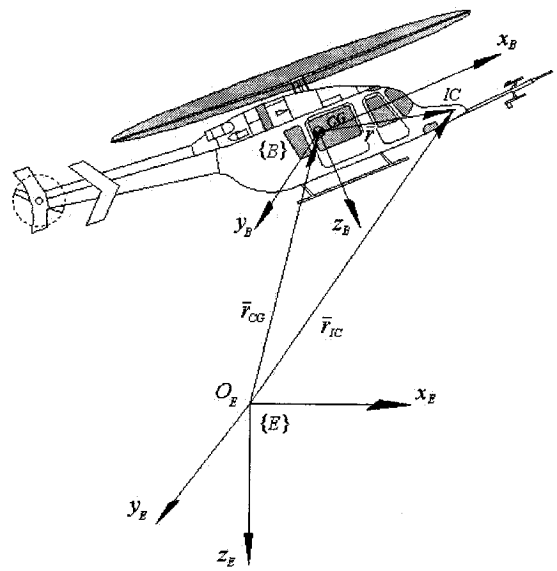


Figure 3-3 A helicopter in Body-fixed and Earth reference systems

The accelerometers located in the *IC* are used to record the accelerations of the helicopter. The components of the accelerations indicated by the accelerometers at the *IC* point are a_{x_m}, a_{y_m}, a_{z_m}. Their expressions are [13]:

$$\left.\begin{aligned} a_{x_m} &= a_{x_{IC}} - g\sin\theta \\ a_{y_m} &= a_{y_{IC}} + g\cos\theta\sin\phi \\ a_{z_m} &= a_{z_{IC}} + g\cos\theta\cos\phi \end{aligned}\right\} \tag{3.1}$$

Therefore, the components of the accelerations at the IC are:

$$\left.\begin{aligned} a_{x_{IC}} &= a_{x_m} + g\sin\theta \\ a_{y_{IC}} &= a_{y_m} - g\cos\theta\sin\phi \\ a_{z_{IC}} &= a_{z_m} - g\cos\theta\cos\phi \end{aligned}\right\} \tag{3.2}$$

Because of the difference between the IC and the CG locations, the accelerations at the IC need to be corrected and translated to the CG position.

According to the assumptions in Chapter 1, the helicopter is treated as a rigid body. Considering the helicopter in the Earth reference system $\{E\}$ shown in Figure 3-3, with its body-fixed frame $\{B\}$ rotating with an angular velocity $\bar{\omega}$ about the Earth reference system $\{E\}$, we have the following equation:

$$\bar{r}_{CG} = \bar{r}_{IC} - \bar{r} \tag{3.3}$$

where $\bar{r}_{CG}$ is the CG position in the frame $\{E\}$;

$\bar{r}_{IC}$ is the IC position in the frame $\{E\}$;

$\bar{r}$ is the IC position relative to the origin of body-fixed frame. Expressed in this frame, the vector $\bar{r}$ takes the form $\bar{r} = (x_{IC}, y_{IC}, z_{IC})^T$.

The velocity of CG is

$$\bar{V}_{CG} = \frac{d\bar{r}_{CG}}{dt}\Big|_E = \frac{d\bar{r}_{IC}}{dt}\Big|_E - \frac{d\bar{r}}{dt}\Big|_E \tag{3.4}$$

By use of equation (1.10), equation (3.4) becomes

$$\bar{V}_{CG} = \frac{d\bar{r}_{IC}}{dt}\Big|_E - \left(\frac{d\bar{r}}{dt}\Big|_B + (\bar{\omega}\times\bar{r})_B\right) \tag{3.5}$$

Considering $\bar{V}_{IC} = \frac{d\bar{r}_{IC}}{dt}\Big|_E$ and $\frac{d\bar{r}}{dt}\Big|_B = 0$ for a rigid body, equation (3.5) is written as

$$\bar{V}_{CG} = \bar{V}_{IC} - (\bar{\omega}\times\bar{r})_B \tag{3.6}$$

where $\bar{V}_{CG}$ is the velocity of the CG measured in the inertial frame $\{E\}$, $\bar{V}_{IC}$ is the velocity of IC measured in the inertial frame $\{E\}$.

$$(\bar{\omega}\times\bar{r})_B = \begin{vmatrix} \bar{i} & \bar{j} & \bar{k} \\ p & q & r \\ x_{IC} & y_{IC} & z_{IC} \end{vmatrix} = (qz_{IC} - ry_{IC})\bar{i} + (rx_{IC} - pz_{IC})\bar{j} + (py_{IC} - qx_{IC})\bar{k} \tag{3.7}$$

Substituting equation (3.7) into (3.6), the components of $\bar{V}_{CG}$ in the body-fixed frame are

$$\left.\begin{aligned} u_{CG} &= u_{IC} - qz_{IC} + ry_{IC} \\ v_{CG} &= v_{IC} - rx_{IC} + pz_{IC} \\ w_{CG} &= w_{IC} - py_{IC} + qx_{IC} \end{aligned}\right\} \tag{3.8}$$

From equation (3.6), we obtain

$$\frac{d\bar{V}_{CG}}{dt}\Big|_E = \frac{d\bar{V}_{IC}}{dt}\Big|_E - \frac{d(\bar{\omega}\times\bar{r})_B}{dt}\Big|_E \tag{3.9}$$

where $\frac{d\bar{V}_{CG}}{dt}\Big|_E = \bar{a}_{CG}$, $\frac{d\bar{V}_{IC}}{dt}\Big|_E = \bar{a}_{IC}$

$$\frac{d(\bar{\omega}\times\bar{r})_B}{dt}\Big|_E = \frac{d(\bar{\omega}\times\bar{r})_B}{dt}\Big|_B + \bar{\omega}\times(\bar{\omega}\times\bar{r}) = \dot{\bar{\omega}}\times\bar{r} + \bar{\omega}\times\dot{\bar{r}} + \bar{\omega}\times(\bar{\omega}\times\bar{r}) \text{ and } \dot{\bar{r}} = 0.$$

We obtain the following expression:

$$\bar{a}_{CG} = \bar{a}_{IC} - \dot{\bar{\omega}}\times\bar{r} - \bar{\omega}\times(\bar{\omega}\times\bar{r}) \tag{3.10}$$

where $\dot{\bar{\omega}} = (\dot{p},\dot{q},\dot{r})^T$ is the rotational acceleration. Substituting equation (3.7) into equation (3.10) gives:

$$\begin{aligned}\bar{\omega}\times(\bar{\omega}\times\bar{r}) = & \left[-(q^2+r^2)x_{IC} + pqy_{IC} + prz_{IC}\right]\bar{i} \\ & + \left[-(p^2+r^2)y_{IC} + qrz_{IC} + pqx_{IC}\right]\bar{j} \\ & + \left[-(p^2+q^2)z_{IC} + prx_{IC} + qry_{IC}\right]\bar{k}\end{aligned}$$

The components of the linear accelerations at the CG ($a_{x_{CG}}, a_{y_{CG}}, a_{z_{CG}}$) are:

$$\left.\begin{aligned} a_{x_{CG}} &= a_{x_{IC}} + (q^2+r^2)x_{IC} - (pq-\dot{r})y_{IC} - (pr+\dot{q})z_{IC} \\ a_{y_{CG}} &= a_{y_{IC}} - (pq+\dot{r})x_{IC} + (p^2+r^2)y_{IC} - (qr-\dot{p})z_{IC} \\ a_{z_{CG}} &= a_{z_{IC}} - (pr-\dot{q})x_{IC} - (qr+\dot{p})y_{IC} + (p^2+q^2)z_{IC} \end{aligned}\right\} \tag{3.11}$$

where $(a_{x_{IC}}, a_{y_{IC}}, a_{z_{IC}})$ is given by equation (3.2).

3.2.2 Transformations of α, β and airspeed from the *IC* to the *CG*

The alpha and beta vanes measure the local flow direction. Figure 3-2 shows the locations of the two vanes. The airflow resulting from the angular velocities and flight path curvature introduces errors in the measured flow angles with respect to the true

angle of attack or sideslip angle. The correction formulas for the angle of attack and sideslip angle from the nose boom to CG are as follows [13]:

$$\alpha_{V_{mp}} = \alpha_{CG} - \frac{x_\alpha}{V_{CG}^2}\left(a_{z_{CG}} - g\cos\theta\cos\phi\right) + \frac{x_\alpha}{V_{CG}}q \tag{3.12}$$

where x_α is the distance between the alpha vane and the aircraft CG in the x_B axis direction. V_{CG} is the true velocity at the CG. $a_{z_{CG}}$ is the acceleration at the CG along the

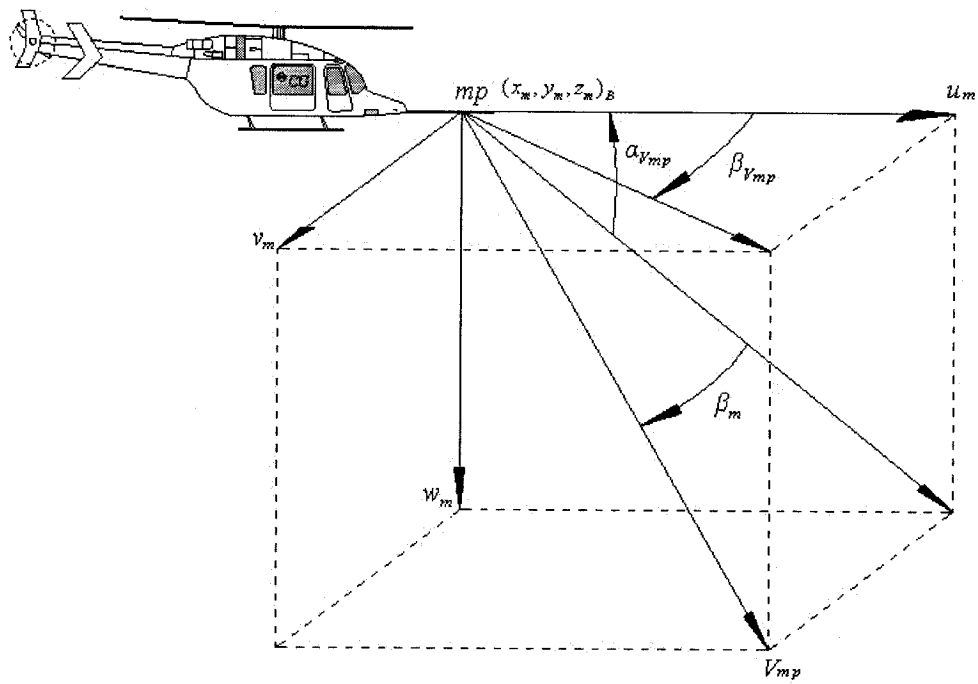


Figure 3-4 The indicated true airspeed V_{mp} at the measured point mp

body axes z_B and q is the pitch rate. $\alpha_{V_{mp}}$ is the measured value of alpha vane as shown in Figure 3-4.

In Figure 3-4, mp denotes the measured point. Its position is expressed

as $(x_m, y_m, z_m)_B$ in the body-fixed frame; u_m, v_m, w_m are the velocities along the body-fixed axis x_B, y_B, z_B, respectively. V_{mp} and β_{Vmp} denote the indicated air speeds (IAS) and the measured value of beta vane at measured point, respectively.

Flight path curvature has a negligible effect in yaw on the sideslip vane. An expression for correcting the indicated sideslip angle by considering corrections for yaw-rate and roll-rate effects is [13]:

$$\beta_{CG} = \beta_m - \frac{x_\beta}{V_{CG}} r + \frac{z_\beta}{V_{CG}} p \tag{3.13}$$

where x_β is the distance between the beta vane and the aircraft CG in the x_B axis direction. z_β is the distance between the beta vane and the aircraft CG in the z axis direction. p and r are the roll-rate and the yaw-rate, respectively.

β_m is the sideslip angle which is different from the measured value (β_{Vmp}) of beta vane, and is defined in Figure 3-4. The real angle of sideslip β_m is determined by following expression,

$$\beta_m = \arctan\left(\tan\beta_{Vmp}\cos\alpha_{Vmp}\right) \tag{3.14}$$

where $\beta_{Vmp} = \tan^{-1}(v_m / u_m)$

By use of the measured airspeed V_{mp}, angle of attack α_{Vmp} and angle of sideslip β_{Vmp}, the longitudinal, lateral and normal speed components at the sensor position are calculated by:

$$\left.\begin{aligned} u_m &= V_{mp}\cos\alpha_{Vmp}\cos\beta_{Vmp} \\ v_m &= V_{mp}\sin\beta_{Vmp} \\ w_m &= V_{mp}\sin\alpha_{V_{mp}}\cos\beta_{Vmp} \end{aligned}\right\}$$

(3.15)

Substituting equation (3.15) into equation (3.8), where (u_m, v_m, w_m) represent (u_{IC}, v_{IC}, w_{IC}) and (x_m, y_m, z_m) represent (x_{IC}, y_{IC}, z_{IC}), the velocity in the body-fixed frame and the True Air Speed (TAS) at the *CG* are:

$$\left.\begin{aligned} u_{CG} &= u_m - qz_m + ry_m \\ v_{CG} &= v_m - rx_m + pz_m \\ w_{CG} &= w_m - py_m + qx_m \end{aligned}\right\} \qquad (3.16)$$

The true air speed (TAS) at the *CG* is

$$TAS = \sqrt{u_{CG}^2 + v_{CG}^2 + w_{CG}^2} \qquad (3.17)$$

In this chapter, we have derived the transformation expressions of accelerations and airspeed from the Instrumentation Center (IC) to the Center of Gravity (CG) and the expression of TAS, these expressions were used by the university team to reorganize the raw flight test data so that all partners involved in the project could use them easily. In next chapter, we will introduce helicopter simulation model validation which was the most labour intensive and time consuming part of the project.

CHAPTER 4

SIMULATION MODEL VALIDATION

The helicopter simulation model validation is to evaluate and approve the simulation model which is rebuilt by use MLE from the flight data. In this chapter, we present the validation requirements according to Advisory Circular AC 120-63 [15] and the Quality Test Guide, introduce the simulation model validation process by use of software POM, analyze the relationship between the initial condition parameters and the simulated outputs, and explain the time consuming nature of validation as currently done.

4.1 General description

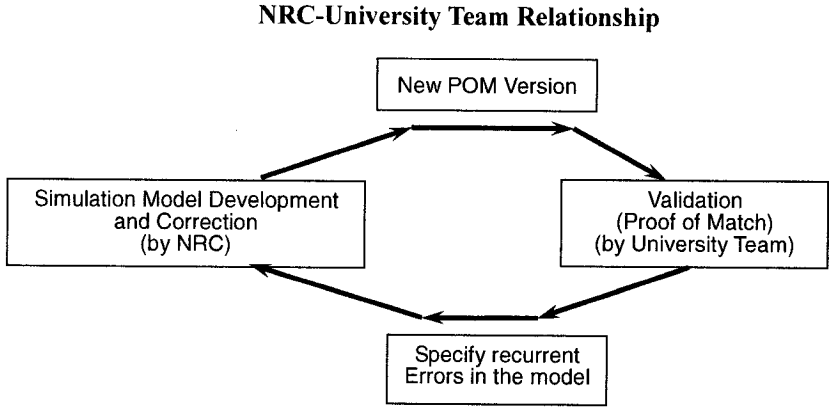


Figure 4-1 Relationship between NRC and University team

The university team worked in conjunction with *National Research Council Canada (NRC)* and *Bell Helicopter Textron Canada Limited (BHTCL).* The relationship between

the team and NRC is shown in Figure 4-1. By use of POM (the Proof of March software), we compare helicopter evolution recorded during flight tests with the results obtained from the model simulation with the same initial conditions and control inputs as the flight test. There are usually differences between the simulated results and the recorded data. These differences are from signals errors and noise recorded by the instrumentation. Due to the fact that the method used to obtain the helicopter model is the Maximum Likelihood Estimation Method (MLE), the simulated results constitute parameter estimations for a certain flight test case. The model validation (called Proof of Match) is required to ensure that the estimated parameters are near the recorded parameters and are located within FAA tolerances [15] depending on the studied flight test case. Following changes in initial conditions such as $p, q, r, \dot{p}, \dot{q}, \dot{r}...$, the differences between the simulated results and the recorded data change due to the fact that the model output depends on the initial conditions. The best possible simulated output is sought by adjusting the initial conditions within reasonable limits. Once the error between the model and flight test output has been minimized, we evaluate the trends in the model errors and communicate our observation to NRC, who corrects the model according to our recommendations.

4.2 Validation requirements

The helicopter identification process needs to reconstruct simulation models by use of the parameters from flight test data in various flight conditions. The validation process is

necessary to use these models to predict flying qualities parameters. In the simulation world, the measure of quality is described as the validation level [2]. Validation level is normally judged with respect to flight test data. If the identification of the stability and control derivatives from the different flight data in flight conditions are correct, i.e. the model structures are correct, the models should agree with the data used for validation in the same flight conditions.

For the helicopter simulator certification for training programs and airman checking under various parts of the Federal Aviation Regulations, the Quality Test Guide (QTG) is used to test the simulator programming and the models evaluated and approved in accordance with Advisory Circular AC 120-63 [15]. Currently, there are three levels of complexity for helicopter simulators: levels B, C and D. In the A/C certification, the simulator requirements are described for qualification of Level B, Level C, and Level D helicopter simulators. The criteria and the guidelines for the helicopter simulator evaluation and qualification are provided. The simulator model of the Bell 427 helicopter must satisfy the highest level D requirements.

The QTG is a list of compulsory requirements described in Table 4.1 for the validation of a helicopter simulator. This table contains all manoeuvres and several configurations of the helicopter during these manoeuvres which cover the helicopter flight envelope.

For the helicopter simulator certification, the software which is written in the Matlab language and used for the identification and the validation of the Bell 427 helicopter has

been developed by NRC, and is called Proof-of-Match (POM). The use of this software will be presented later. A detailed description of its use is described in reference [14].

All the simulation flight results from Table 4.1 should be matched with the flight recorded data within the tolerances stated in the Appendix 2 of AC 120-63 [15]. Here Parts of the Appendix 2 are presented in Table 4.2 for various flight conditions:

Table 4.1 Quality Test Guide (1 of 2)

Code	Manoeuvre	GW	CG	Hd (ft)	IAS (knots)
1a	**Engine Assessment**				
1a1a	**Start to Steady State Idle**				
	Rotor Brake Used - Eng #1	-	-	0	-
	Rotor Brake Used - Eng #2	-	-	0	-
	Rotor Brake NOT Used - Eng #1	-	-	0	
	Rotor Brake NOT Used - Eng #2	-	-	0	-
1a1b	**Steady State Idle to Operation RPM**				
	Rotor Brake Used - Eng #1	-	-	0	-
	Rotor Brake Used - Eng #2	-	-	0	-
	Rotor Brake NOT Used - Eng #1	-	-	0	
	Rotor Brake NOT Used - Eng #2	-	-	0	-
1a2	**Power Turbine Speed Trim (Can use data from other climb/descent tests)**				
	Power Up Collective Step Inputs.	-	-	0	-
	Power Down Collective Step Inputs.	-	-	0	-
1a3	**Engine & Rotor Speed (Climb/Descent)**				
	Engine & Rotor Speed - Climb	-	-	3000	
	Engine & Rotor Speed - Descent	-	-	3000	
1d	**Hover Performance**				
	IGE	Light	-	IGE	0
	IGE	Heavy	-	IGE	0
	OGE	Light	-	OGE	0
	OGE	Heavy	-	OGE	0
1e	**Vertical Climb**				
	Heavy Weight	Heavy	-	OGE	-
	Light Weight	Light	-	OGE	-
1f	**Level Flight (Two gross weigh/CG combination)**				
	Weight/CG Combination #1 - Cruise	Light	Aft	3000	60
	Weight/CG Combination #2 - Cruise	Heavy	Fwd	3000	80
1g	**Climb Performance (Two gross weight/CG combination)**				
	All Engines Operative #1	Light	Aft	1500	60
	All Engines Operative #2	Heavy	Aft	3350	115
	One Engine Inoperative #1	-	-	6000	
	One Engine Inoperative #2	-	-	6000	
1h1	**Descent Performance (Two gross weight/CG combination)**				
	Weight/CG combination #1	Light	Aft	3000	70
	Weight/CG combination #2	Heavy	Fwd	3000	50
1h2	**Autorotation Performance (Two gross weights)**				
	Heavy				
	Light				

Table 4.1 Quality Test Guide (continue 2 of 2)

Code	Manoeuvre	GW	CG	Hd (ft)	IAS (knots)
1j	**Landing**				
1j1	All Engines Operative	-	-		
1j2	OEI – Cat A Completed	-	-		
2b1	**Low Speed Controllability (Several speed increments to translational limit and 45 knots)**				
	Trimmed Flight Conditions – Forward	-	-	IGE	
	Trimmed Flight Conditions – Sideward	-	-	IGE	
	Trimmed Flight Conditions – Rearward	-	-	IGE	
2b2	**Critical Azimuth (Three relative wind directions)**				
	Direction #1	-	-	Hover	
	Direction #2	-	-	Hover	
	Direction #3	-	-	Hover	
2b3	**Control Response**				
2b3a	Longitudinal	-	-	Hover	
2b3b	Lateral	-	-	Hover	
2b3c	Directional	-	-	Hover	
2b3d	Vertical	-	-	Hover	
2c	**Longitudinal Handling Qualities**				
2c1	**Control Response (F/A Step inputs and Collective)**				
	Control Response – F/A	Light	Aft	3000	117
	Control Response – Collective	Heavy	Fwd	3000	60
2c2	**Static Stability (Minimum 2 speeds on each side of trim)**				
	Static Stability – Cruise/Climb	Heavy	Aft	3000	91
	Static Stability – Cruise/Climb	Heavy	Fwd	3000	103
2c2	**Static Stability - Autorotation**	-	-	3000	
2c3	**Dynamic Stability (Cruise or Climb. Minimum 2 speeds on each side of trim)**				
2c3a	Dynamic Stability – Long period	-	-	3000	
2c3b	Dynamic Stability - Short Period- Speed #1	-	-	3000	
2c3b	Dynamic Stability - Short period- Speed #2	-	-	3000	
2c4	**Manoeuvring Stability (Minimum 2 speeds on each side of trim)**				
	Manoeuvring Stability - Speed 1 (30 deg)	Heavy	Aft	3000	60
	Manoeuvring Stability - Speed 2 (30 deg)	Heavy	Aft	3000	117
	Manoeuvring Stability - Speed 1 (45 deg)	Heavy	Aft	3000	60
	Manoeuvring Stability - Speed 2 (45 deg)	Heavy	Aft	3000	117
2d	**Lateral/Directional Handling Qualities**				
2d1	**Control Response**				
2d1a	Control Response - Lateral	Light	Aft	3000	60
2d1b	Control Response - Directional Speed 1	Light	Aft	3000	117
2d1b	Control Response - Directional Speed 2	Heavy	Aft	3000	60
2d2	**Directional Static Stability**				
	Directional Static Stability - Climb/Cruise	Heavy	Aft	3000	60
	Directional Static Stability - Climb/Cruise	Heavy	Fwd	3000	117
	Directional Static Stability - Descent	Heavy	Aft	3000	102
2d3a	**Dynamic Stability, Lateral/Directional Oscillations**				
	Dynamic Stability - Climb/Cruise Speed 1	Light	Aft	3000	60
	Dynamic Stability - Climb/Cruise Speed 2	Light	Aft	3000	115
2d3b	**Spiral Stability (Cyclic Only turns Left and Right turns until +/- 30° bank angle)**				
	Spiral Stability - Cruise/Climb Left	Heavy	Aft	3000	60
	Spiral Stability - Cruise/Climb Right	Heavy	Aft	3000	60

Table 4.2 Table of validation tests – tolerances and flight conditions

Code	Tests	Tolerance	Flight Conditions and Comments
1c	**Takeoff (All engines and OEI)**	Airspeed ±3 kt Altitude ±20 ft (6.1m) Torque ±3% Rotor Speed ±1.5% Vertical Velocity ±100 fpm (0.5 m/sec) or 10% Pitch Attitude ±1.5 deg Bank Attitude ± 2 deg Heading ± 2 deg Longitudinal Control Position ± 10% Lateral Control Position ± 10% Directional Control Position ± 10% Collective Control Position ±10%	Ground/Takeoff and Initial Segment of Flight. Time history of takeoff flight path as appropriate to helicopter model simulated. Record data to at least 200ft (61m) AGL.
1d	**Hover Performance**	Torque ±3% Pitch Attitude, Bank Attitude ±1.5 deg Longitudinal Control Position ±5% Lateral Control Position ±5% Directional Control Position ±5% Collective Control Position ±5%	In Ground Effect (IGE) Out of Ground Effect (OGE) Light/Heavy Gross Weights Snapshot Test
1e	**Vertical Climb**	Vertical Velocity ±100 fpm (0.5 m/sec) or 10% Directional Control Position ±5% Collective Control Position ±5%	From OGE Hover. Light/Heavy Gross Weights Snapshot Test
1i	**Autorotational Entry**	Rotor speed ± 3% Pitch Attitude ± 2 deg Roll Attitude ± 3 deg Yaw Attitude ± 5 deg Airspeed ± 5 kt Vertical Velocity ± 200 fpm (1 m/sec) or 10%	Cruise or Climb Time history of vehicle response to a rapid throttle reduction to idle.
2d1	**Lateral and Directional Handling Qualities** **(a) Control Response Lateral**	Roll Rate ± 10% or ± 3 deg/sec Roll Attitude Change ± 10% or 3 deg	Cruise Augmentation On/Off Step control input Off axis response must show correct trend for unaugmented cases.
2d1	**(b) Control Response Directional**	Yaw Rate ± 10% or ± 2 deg/sec Yaw Attitude Change ± 10% or ± 2 deg	Cruise Augmentation On/Off Step control input Off axis response must show correct trend for unaugmented cases.
2d2	**Directional Static Stability**	Lateral Control Position ± 10% of change from trim Roll Attitude ± 1.5 deg Directional Control Position ± 10% of change from trim Longitudinal Control Position ± 10% of change from trim Vertical Velocity ± 100 fpm (0.5m/sec) or 10%	Cruise or Climb/Descent Augmentation On/Off Steady heading sideslip Minimum of two sideslip angles on either side of the trim point. Snapshot test.
2d3	**Dynamic Lateral and Directional Stability** **(a) Lateral-Directional Oscillations**	± 0.5 sec or 10% of Period ± 10% of time to ½ or Double Amplitude or ± .02 of Damping Ratio ± 20% or ± 1 sec of Time Difference Between Peaks of Bank and Sideslip	Cruise or Climb Augmentation On/Off Two Airspeeds. Excite with cyclic or pedal doublet. Test should include six full cycles (12 overshoots after input completed) or that sufficient to determine time to ½ or Double Amplitude whichever is less. For non-periodic response, time history should be matched
2d3	**(b) Spiral Stability**	Correct Trend, ± 2 deg Bank or 10% in 20 sec	Cruise or Climb Augmentation On/Off Time history of release from pedal only or cyclic only turns in both directions.

4.3 Validation introduction

The helicopter is modeled by the dynamic equations (2.37):

$$\dot{\bar{x}}(t) = A\bar{x}(t) + B\bar{u}(t) + \bar{f}(t) \tag{4.1}$$

where $\bar{x}(t) = [u, w, q, \theta, v, p, \phi, r]^T$.

Because $\bar{f}(t)$ is the turbulence, it cannot be presented by an analytical expression, theoretically, the solution of the simulation model can be only written as:

$$\bar{x}(t) = \bar{x}_0 + \int_{t_0}^{t} \left(A\bar{x}(t) + B\bar{u}(t) \right) dt \tag{4.2}$$

where $\bar{x}_0 = \bar{x}(t_0)$ is the initial condition vector at time t_0. Clearly $\bar{x}_0$ affects $\bar{x}(t)$. If the stability derivatives in matrix A and control derivatives in matrix B are correctly identified from the flight data in the flight conditions by the identification method, and $\bar{x}(t_0)$ is correctly given, $\bar{x}(t)$ in equation (4.2) will match the recorded data in the same flight conditions. Considering the noise of signals and unexpected perturbations, the error between the integration values $\bar{x}(t)$ and flight test data should be within the tolerances in Table 4.2. In other words, if the matrices A, B are correct, we can find a reasonable $\bar{x}(t_0)$ to match the $\bar{x}(t)$ with the flight data, considering the turbulence the error should be within the tolerances (i.e. FAA tolerance).

The validation process of the helicopter dynamic model is to observe whether the simulation output plots are within the tolerances in Table 4.2 by changing the initial condition vector components of $\bar{x}(t_0)$; if yes, the model is evaluated 'true', otherwise

the model is valuated 'false' and needs to be corrected. The reasons why the model may be false are the following:

(1) mistake in programming;

(2) poorly estimated stability and control derivatives;

(3) unexpected non-linearities in the helicopter dynamics;

(4) poor results given by trim function.

We use the POM software to validate the helicopter dynamic model. The use of the software POM will be introduced later. In order to validate this model for different flight conditions, we are allowed to edit a portion of the initial parameters dependent on the selection of trim (control input) and subtrim (stability derivatives) function combinations. When a certain trim function is selected, the POM operator chooses a set of parameters whose inputs can be controlled by adjustments of initial conditions. When a certain subtrim function is selected, the operator chooses the derivative values of the initial conditions to be zero or non-zero, while all the others parameters are left unconstrained.

The trim and the subtrim functions are selected from the following list:

(a) Trim function:

'TrimRate' edit: p, q, r

'TrimAccel' edit: p, q, r, A_x, A_y, A_z

'TrimAngAccel' edit: $\dot{p}, \dot{q}, \dot{r}$

'TrimSideslip' edit: p, q, r, β

'TrimSideslipM' edit: $p, q, r, \beta, Mach$

'TrimRoll' edit: p, q, r, ϕ

(b) Subtrim function:

'subTrimRate' initial condition imposed $\dot{p}, \dot{q}, \dot{r} = 0$

'subTrimVelRate' initial condition imposed $\dot{u}, \dot{v}, \dot{w}, \dot{p}, \dot{q}, \dot{r} = 0$

'subTrimClimb' initial condition imposed $\dot{u}, \dot{v}, \dot{w}, \dot{p}, \dot{q}, \dot{r}$,ROC = 0

'subTrimClimbAngAccel' initial condition imposed u, v, w, ROC = 0, edit: $\dot{p}, \dot{q}, \dot{r}$

'subTrimRateROC' initial condition imposed $\dot{p}, \dot{q}, \dot{r} = 0$, edit: ROC ~= 0

'subTrimVelRateROC' initial condition imposed $\dot{u}, \dot{v}, \dot{w}, \dot{p}, \dot{q}, \dot{r} = 0$ edit: ROC ~= 0

'subTrimClimbROC' initial condition imposed $\dot{u}, \dot{v}, \dot{w}, \dot{p}, \dot{q}, \dot{r} = 0$ edit: ROC ~=0

From the above list, we can see that there are 13 initial parameters which can be edited for validation: $p_0, q_0, r_0, a_{x0}, a_{y0}, a_{z0}, \phi_0, \beta_0, Mach_0, ROC_0, \dot{p}_0, \dot{q}_0, \dot{r}_0$. The POM operator is allowed to edit (or change) different groups of the thirteen parameters depending on the different combinations of trim and subtrim functions selected.

The POM software outputs 16 parameter plots and shows four input parameter plots; the output parameter plots are the time history plots of p, q, r, ϕ, θ, ψ, α, β, $Mach$, u, v, w, a_x, a_y, a_z, H. The changes of the initial parameters at time t_0 cause changes of the sixteen output parameters. p_0, q_0, r_0 are angular velocities. They are related to the $\dot{\phi}_0, \dot{\theta}_0, \dot{\psi}_0$ given by equation (1.4). Therefore, the changes of p_0, q_0, r_0 give rise to changes in the output parameters p, q, r ϕ, θ, ψ.

$\dot{p}_0, \dot{q}_0, \dot{r}_0$ are angular accelerations, and are related to the moment and the angular velocities p_0, q_0, r_0. Therefore, the $\dot{p}_0, \dot{q}_0, \dot{r}_0$ changes cause changes in the output parameters p, q, r ϕ, θ, ψ. The linear accelerations a_{x0}, a_{y0}, a_{z0}, are related to the force and the linear velocities $u_0, v_0, w_0, p_0, q_0, r_0$ from equation (1.16).

$\beta_0, Mach_0, \mathrm{ROC}_0$ can be obtained by substituting u_0, v_0, w_0, θ_0, α_0 into the following equations (4.3), (4.4) and (4.5):

$$\beta = \tan^{-1}(v/\sqrt{u^2+w^2}) \tag{4.3}$$

$$Mach = \sqrt{u^2+v^2+w^2}/a \tag{4.4}$$

$$\mathrm{ROC} = \sqrt{u^2+v^2+w^2}\tan(\alpha-\theta) \tag{4.5}$$

where a is the speed of sound. The angle of attack α is determined by the equation $\alpha = \tan^{-1}(w/u)$.

β_0 and $Mach_0$ are related to the linear velocities in the body-fixed frame. ROC_0 is related to linear velocities and pitch angle θ_0 and H. The roll angle ϕ_0 is related to the gravity direction as well as to p_0, q_0, r_0.

We see that a change in one of initial parameter may cause changes in one or more output parameters.

Theoretically, any trim function can be combined with any subtrim function, but the results may be significantly different, and sometimes the results between trim and subtrim functions are very far from the real helicopter behaviour. For this reason, the POM operator should choose the best combination of trim and sub-trim functions which

gives the most appropriate validation of the theoretical model results with the helicopter flight test results.

4.4 Use of the POM software

POM was developed by NRC for helicopter simulation in Matlab, and includes a flight model. A POM operator cannot read the helicopter dynamics model code due to NRC copyright protection, therefore the model is a black-box for the operator. On the other hand, the operator can read and change the initial conditions and the parameters tolerances required for the parameters. The initial conditions are stored in a Test Definition File (TDF) and the tolerances for each flight condition parameters are stored in a Graphic Definition File (GDF). Figure 4-2 shows TDF and GDF options files. These

TDF Option

```
                26-Jan-2006 13:22:24 POM7.2
        Manoeuver: landing-OEI-cat A/B completed (1j2)
              Model selected: Up-and-Away (UAA)
                        No Ground Effect
DetaFile        ='t904r19_m.dat';
Manoeuver       ='ij2' (from QTG_check xls)
StartTime       =4.5  [sec]
endTime         = 6.5  [sec]
TRIMF           ='TrimAngAccel'         -> edit: p,q,r,PDOT,QDOT,RDOT
subTRIMf        = 'subTrimRateROC';   -> p/q/rdot = 0     edit: ROC ~= 0
ROLL_RATE           =     0.187;          [deg/s]
PITCH_RATE          =     1.820;          [deg/s]
YAW_RATE            =    -1.458;          [deg/s]
X_ACCEL             =     2.855;          [ft/s2]
Y_ACCEL             =    -0.789;          [ft/s2]
Z_ACCEL             =   -35.209;          [ft/s2]
U_VEL               =   116.239;          [ft/s]
V_VEL               =    -0.702;          [ft/s]
W_VEL               =    13.672;          [ft/s]
DENSITY_ALTITUDE    =  2313.085;          [ft]
ROLL_ANGLE          =    -0.901;          [deg]
SIDESLIP_ANGLE      =    -0.344;          [deg]
HEADING_ANGLE       =    -2.049;          [deg]
ANGLE_OF_ATTACK     =     6.708;          [deg]
PITCH_ANGLE         =     5.669;          [deg]
TRUE_AIRSPEED       =   117.042;          [ft/s]
MACH                =     0.105;          [-]
RATE_OF_CLIMB       =  -127.386;          [ft/min]
PDOT                =     0.0;            [deg/s2]
QDOT                =     0.0;            [deg/s2]
RDOT                =     0.0;            [deg/s2]
```

GDF Option

maxControl	= 100	[deg]
Long	= 10	[%] (max/min)
Coll	= 10	[%] (max/min)
Lat	= 10	[%] (max/min)
Ped	= 10	[%] (max/min)
P_rate	= 2	[deg/s] (max/min)
Q_rate	= 2	[deg/s] (max/min)
R_rate	= 2	[deg/s] (max/min)
Phi_att	= 1.5	[deg] (max/min)
Theta_att	= 1.5	[deg] (max/min)
Psi_att	= 1.5	[deg] (max/min)
Alpha_aoa	= 2	[deg] (max/min)
Beta_roll	= 2	[deg] (max/min)
Mach	= 0.003	[-] (max/min)
U_vel	= 3	[ft/s] (max/min)
V_vel	= 3	[ft/s] (max/min)
W_vel	= 3	[ft/s] (max/min)
Ax	= 3	[ft/s2] (max/min)
Ay	= 3	[ft/s2] (max/min)
Az	= 3	[ft/s2] (max/min)
Alt	= 20	[ft] (max/min)
EngTorque	= 3	[%] (max/min)

Figure 4-2 TDF input and GDF input

files specify the manoeuver type, the simulation time interval, the takeoff model and the sub model (no ground effects), the data files and so on.

Both TDF and GDF files are automatically generated by programs independent from the main POM software. Following the generation of files, the operator can edit manually the initial condition values.

'TRIMf' and 'subTRIMf' are the functions used to trim the aircraft and to find all parameter initial conditions (see reference [14]). The following section shows how the POM software is used for one case. In this case, firstly, a POM operator decides the best combination of trim and sub-trim conditions. The combination of the functions TrimAngAccel and subTrimRateROC is selected. This selection gives the operator the possibility to arbitrarily edit the initial conditions of $p, q, r, \dot{p}, \dot{q}, \dot{r}$, ROC (Rate Of Climb).

TDF Option

26-Jan-2006 13:22:24 POM7.2
Manoeuver: landing-OEI-cat A/B completed (1j2)
Model selected: Up-and-Away (UAA)
No Ground Effect

DetaFile	='t904r19_m.dat';		
Manoeuver	='ij2' (from QTG_check xls)		
StartTime	=4.5 [sec]		
endTime	= 6.5 [sec]		
TRIMF	='TrimAngAccel'	-> edit: p,q,r,PDOT,QDOT,RDOT	
subTRIMf	= 'subTrimRateROC';	-> p/q/rdot = 0	edit: ROC ~= 0
ROLL_RATE	= 0.187;		[deg/s]
PITCH_RATE	= 1.820;		[deg/s]
YAW_RATE	= -1.458;		[deg/s]
X_ACCEL	= 2.855;		[ft/s2]
Y_ACCEL	= -0.789;		[ft/s2]
Z_ACCEL	= -35.209;		[ft/s2]
U_VEL	= 116.239;		[ft/s]
V_VEL	= -0.702;		[ft/s]
W_VEL	= 13.672;		[ft/s]
DENSITY_ALTITUDE	= 2313.085;		[ft]
ROLL_ANGLE	= -0.901;		[deg]
SIDESLIP_ANGLE	= -0.344;		[deg]
HEADING_ANGLE	= -2.049;		[deg]
ANGLE_OF_ATTACK	= 6.708;		[deg]
PITCH_ANGLE	= 5.669;		[deg]
TRUE_AIRSPEED	= 117.042;		[ft/s]
MACH	= 0.105;		[-]
RATE_OF_CLIMB	= -127.386;		[ft/min]
PDOT	= 0.0;		[deg/s2]
QDOT	= 0.0;		[deg/s2]
RDOT	= 0.0;		[deg/s2]

(a) Before changes in initial conditions

TDF Option

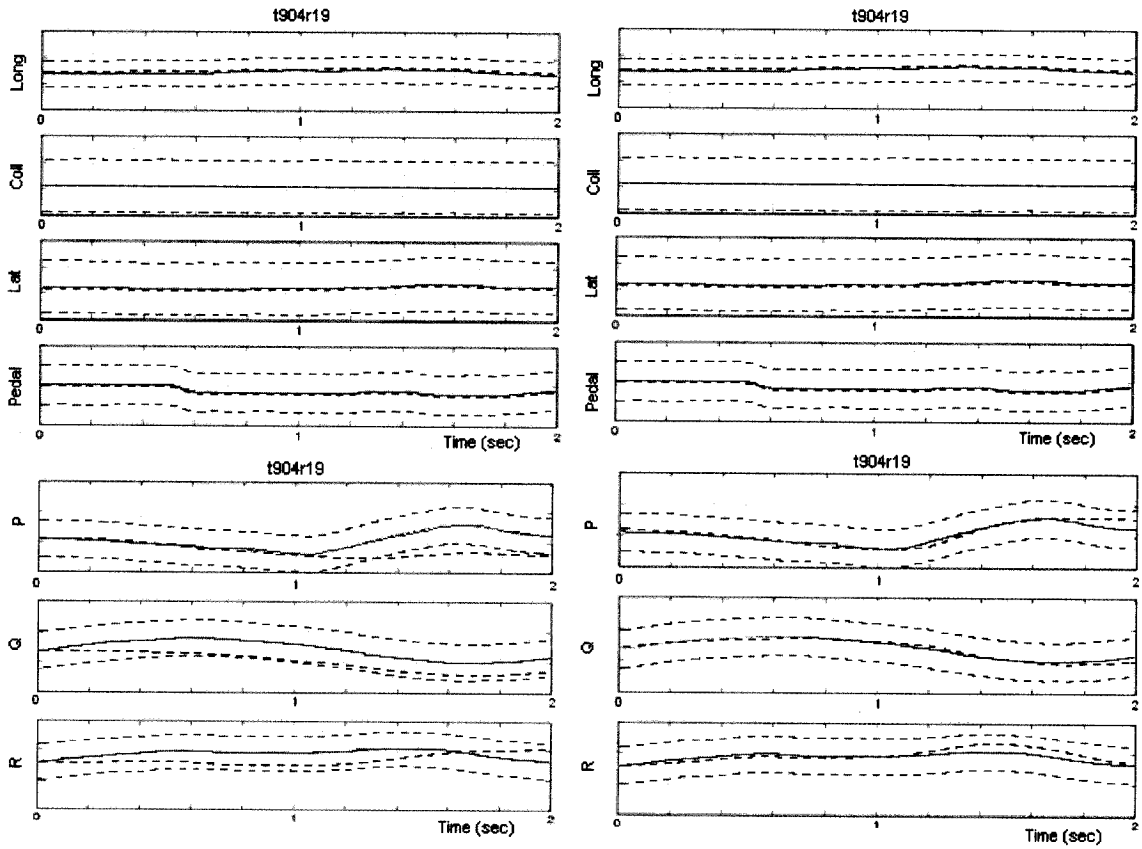
26-Jan-2006 18:02:52 POM7.2
Manoeuver: landing-OEI-cat A/B completed (1j2)
Model selected: Up-and-Away (UAA)
No Ground Effect

DetaFile	='t904r19_m.dat';		
Manoeuver	='ij2' (from QTG_check xls)		
StartTime	=4.5 [sec]		
endTime	= 6.5 [sec]		
TRIMF	='TrimAngAccel'	-> edit: p,q,r,PDOT,QDOT,RDOT	
subTRIMf	= 'subTrimRateROC';	-> p/q/rdot = 0	edit: ROC ~= 0
ROLL_RATE	= 0.187 + 0.42;		[deg/s]
PITCH_RATE	= 1.820 + 0.53;		[deg/s]
YAW_RATE	= -1.458 + 0.39;		[deg/s]
X_ACCEL	= 2.855;		[ft/s2]
Y_ACCEL	= -0.789;		[ft/s2]
Z_ACCEL	= -35.209;		[ft/s2]
U_VEL	= 116.239;		[ft/s]
V_VEL	= -0.702;		[ft/s]
W_VEL	= 13.672;		[ft/s]
DENSITY_ALTITUDE	= 2313.085;		[ft]
ROLL_ANGLE	= -0.901;		[deg]
SIDESLIP_ANGLE	= -0.344;		[deg]
HEADING_ANGLE	= -2.049;		[deg]
ANGLE_OF_ATTACK	= 6.708;		[deg]
PITCH_ANGLE	= 5.669;		[deg]
TRUE_AIRSPEED	= 117.042;		[ft/s]
MACH	= 0.105;		[-]
RATE_OF_CLIMB	= -127.386+150;		[ft/min]
PDOT	= 0.0 + 2.5;		[deg/s2]
QDOT	= 0.0 + 1.5;		[deg/s2]
RDOT	= 0.0 - 0.5;		[deg/s2]

(b) After changes in initial conditions

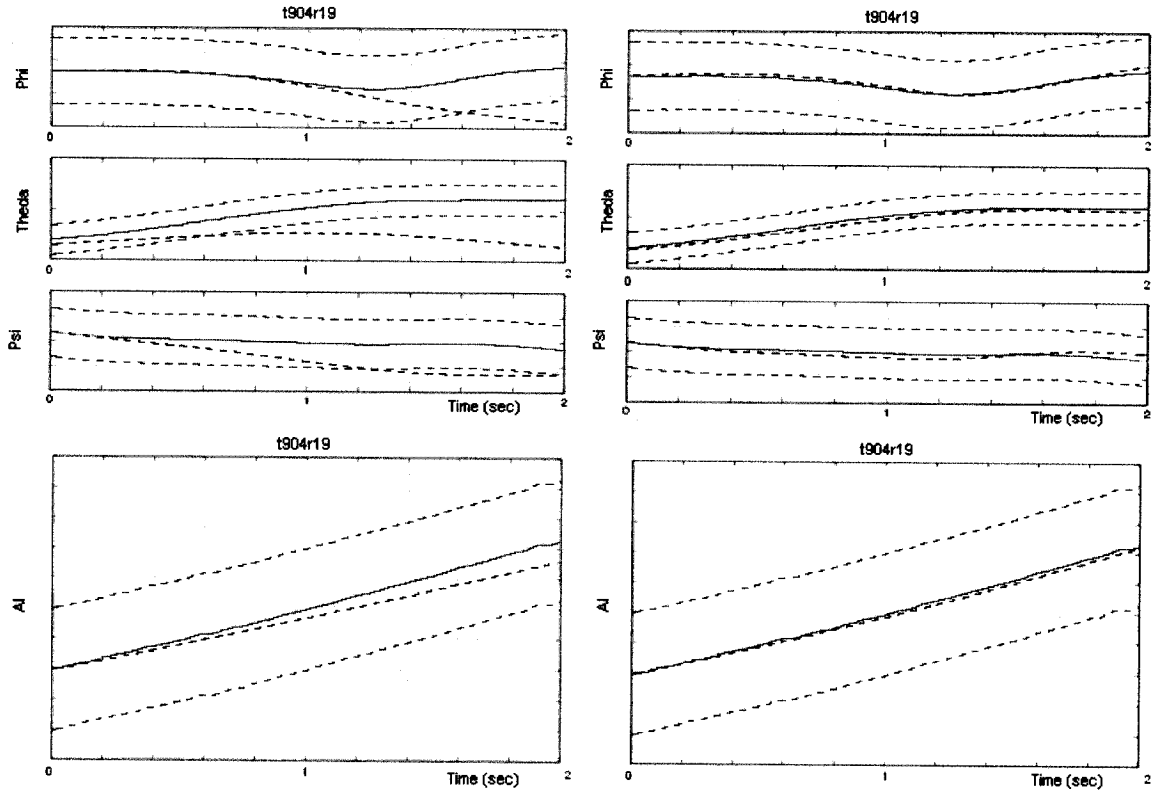
Figure 4-3 Initial condition changes (before and after)

Figures 4-3 shows the comparison of initial conditions. Next, following changes in the initial conditions, the operator monitors the output plots. Initial conditions are modified until the differences between the simulation results and the recorded data are minimum. It is required that the simulation output plots (blue colour) are within tolerances defined by the red dashed lines around flight data (green colour) in Figures 4-4, and 4-5. Time history plots are show before (a) and after (b) initial conditions adjustments for *Pedal*, *Lat*, *Coll*, *Long*, p, q, r (Figure 4-4) and ϕ, θ, ψ, H (Figure 4-5).



(a) Before changes in initial conditions (b) After changes in initial conditions

Figure 4-4 Results comparison (A) before and after changes in initial conditions



(a) Before changes in initial conditions (b) After changes in initial conditions

Figure 4-5 Result comparison (B) before and after changes in initial conditions

Finally, the operator feeds back the results to NRC, who corrects the flight model and then provides the new POM version to the operator.

A flow chart representation of the validation procedure by use of POM is shown in Figure 4-6. We see that the validation process is a trial and error process. This manual trial and error process is time consuming due to the large amount of flight test data in various manoeuvre configurations. In this project, almost 600 flight test cases had to be analyzed by the university team in accordance with the advisory circular FAA AC 120-63 requirements. Usually, the validation of one flight test case takes 4-6 hours or more.

For this reason, the need to develop an automatic POM software to carry out the POM operator's work instead of the POM manual work became evident.

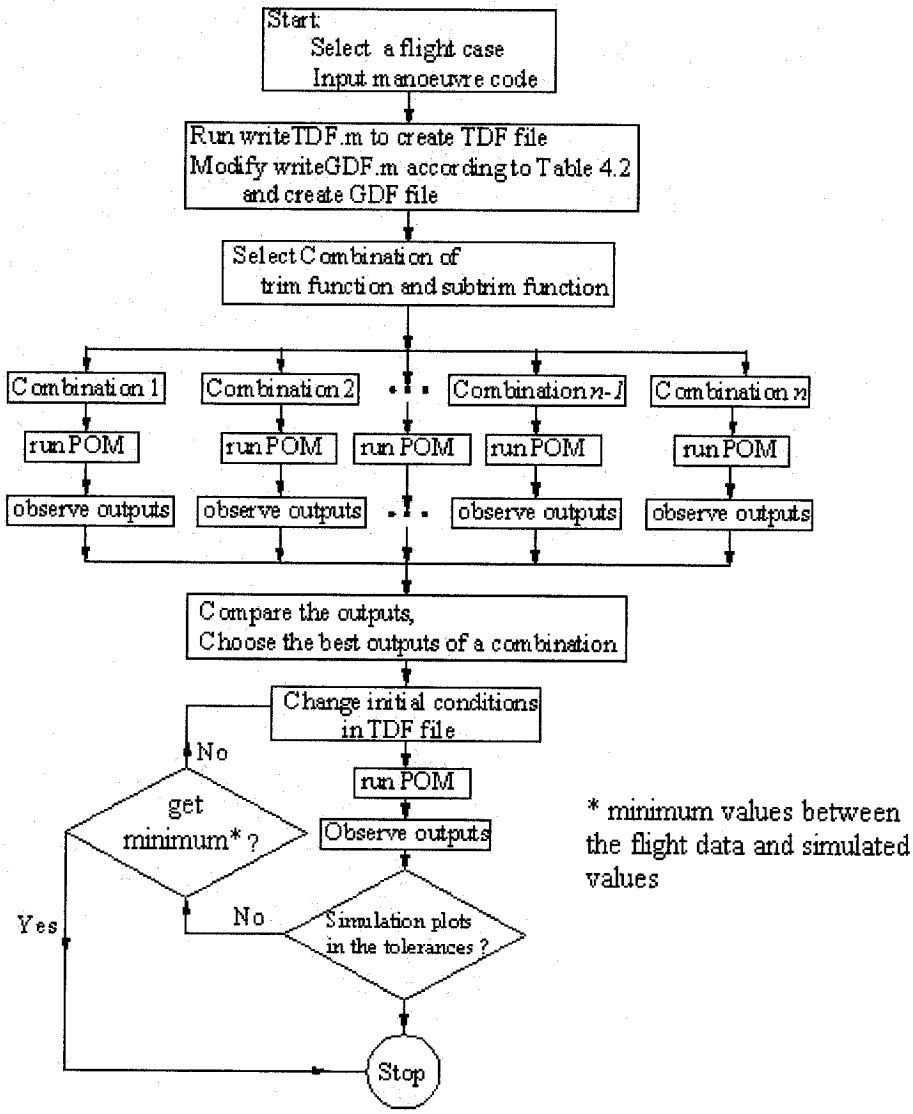


Figure 4-6 Validation process by use of POM

Table 4.3 shows analyzed flight test cases. Using POM (manually), the total time of the 553 flight case validation took about 3000 hours (about 375 working days one person)

The simulation model validation in various flight conditions is described by Popov in his masters thesis [14]. This validation can be described as the output optimization with changes in the initial conditions. In the next chapter, we introduce optimization problems including multi-objective function, the selection of weights.

Table 4.3 Flight tests

AC chapter	Number of flight tests
Longitudinal Handling Qualities	172
Lateral and Directional Handling Qualities	115
Level Flight	59
Descent Performance	11
Climb Performance	14
Hover	74
Low Speed Handling Qualities	50
Autorotation Entry	7
Turns on One Control	24
Vertical Climb	3
Take-Off Flight	18
Landing	6
Total flight tests	***553***

CHAPTER 5

DESCRIPTION OF OPTIMIZATION PROBLEM

According to the description in the section 4.4, the validation process is to manually find the minimum differences between the results obtained by simulation and the recorded flight data by changing the initial conditions. In this chapter, we discuss optimization problems, objective functions, weights of a multi-objective function, In order to formulate the validation process as a multi-objective optimization problem with constraints.

5.1 Optimization problem

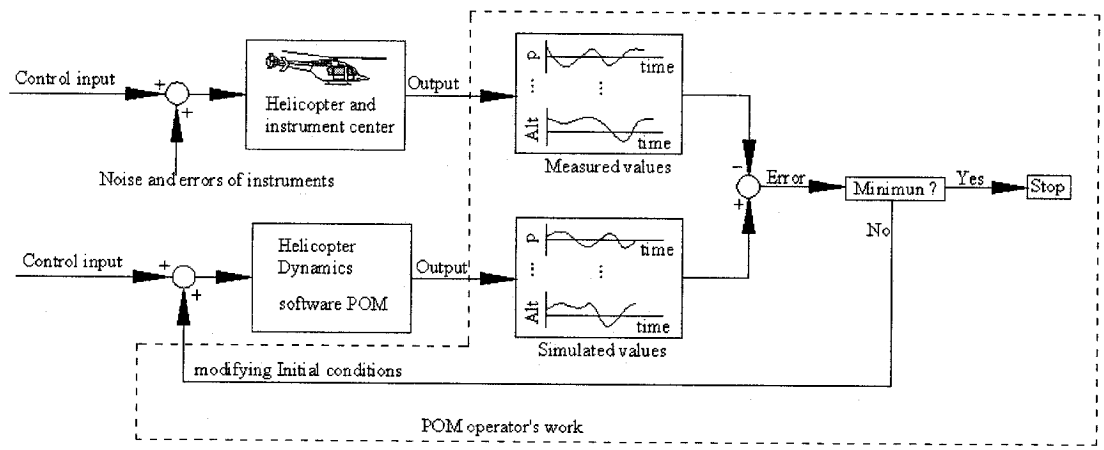


Figure 5-1 The work of a POM operator

The POM operator's work can be expressed by the block graph shown in Figure 5-1. It can also be described as a general mathematical programming problem, which can be

formulated as a multi-objective optimization problem. This mathematical problem can be described by the use of the following objective function, in which the optimal value (minimum) of a set of function (cost function) of multi-variables (initial conditions) should be found and is subject to given tolerances and helicopter dynamics.

5.2 General multi-objective optimization problem

The form of mathematical expression for a general multi-objective optimization problem is

$$\min_{x \in C} F(\bar{x}) = \min_{x \in C} \left[f_1(\bar{x}),\ f_2(\bar{x}),\ \ldots,\ f_n(\bar{x}) \right]^T \tag{5.1}$$

where, $C = \left\{ x : h_k(\bar{x}) = 0,\ k = 1,2,\ldots K,\ \text{and}\ \ g_j(\bar{x}) \le 0,\ j = 1,2,\ldots J \right\}$, $g_j(\bar{x}) \le 0, j = 1,2,\ldots,J$ are inequality constraints, $h_k(\bar{x}) = 0, k = 1,2,\ldots,K$ are equality constraints. K is the number of equality constraints, J is the number of inequality constraints, and n is the number of objective functions. If $n = 1$, our problem becomes a single-objective optimization (SOO) problem and if $n \ge 2$, it becomes a multi-objective optimization (MOO) problem. $\bar{x}$ is a vector of decision variables and represents the optimisation parameter vector. $F(\bar{x})$ is an objective function vector. Each objective function $f_i(\bar{x})$ is an evaluation function representing a scalar cost function.

5.2.1 Single-objective optimization problem cost function

Many different cost functions can be used for different types of problems. The most commonly used cost functions are: *Least Squares* (*LS*) and *Normalized Correlation* (*NC*), *Woods* (*W*), *Correlation Ratio* (*CR*), *Mutual Information* (*MI*), *Normalized Mutual Information* (*NMI*). These functions are defined mathematically in Table 5.1[16].

Table 5.1 Definition of Cost Function

Cost function	Definition	Minimum	Maximum
C^{LS}	$N=\sum(Y-X)^2$	0	∞
C^{NC}	$\frac{\sum(X\cdot Y)}{\sqrt{\sum X^2}\sqrt{\sum Y^2}}$	-1	1
C^{W}	$\sum_k \frac{n_k}{N}\frac{\sqrt{Var(Y_k)}}{\mu(Y_k)}$	0	∞
C^{CR}	$\frac{1}{Var(Y)}\sum_k \frac{n_k}{N} Var(Y_k)$	0	1
C^{MI}	$H(X,Y)-H(X)-H(Y)$	- ∞	0
C^{NMI}	$\frac{H(X,Y)}{H(X)+H(Y)}$	0	1

In this thesis, we choose the *Least Squares* (*LS*) Cost Function C^{LS}. The Least Squares method assumes that the best-fit curve of a given type is the curve that has the minimal sum of the deviations squared (*least square error*) from a given data set. Suppose that the data points are (x_1,y_1), (x_2,y_2),..., where x is the independent variable and y is the dependent variable. The fitted curve $f(x)$ has the deviation (error) d from

each data point, i.e., $d_1 = y_1 - f(x_1)$, $d_2 = y_2 - f(x_2)$,..., $d_n = y_n - f(x_n)$. According to the Least Squares method, the best fitting curve has the following property:

$$N = d_1^2 + d_2^2 + ... + d_n^2 = \sum_{i=1}^{n} d_i^2 = \sum_{i=1}^{n} [(y_i - f(x_i)]^2 = \text{a minimum.} \tag{5.2}$$

Usually, N in equation (5.2) is called a distance or norm function.

5.2.2 Utility function of a multi-objective optimization problem

A multi-objective optimization (MOO) problem is solved in a similar manner to the single-objective optimization (SOO) problem described above. In a SOO problem, the idea is to find a set of variable values which, when subject to a number of constraints, yield an optimum value of the objective (or cost) function. In a MOO problem, we need to find the values for the variables which optimize simultaneously many objective functions. Typically, for MOO problems, there is no single global solution and it is often necessary to determine a set of points which all fit a predetermined optimization definition. In this manner, the solution is chosen from a so-called Pareto optimal set [17] which yields an infinite set of solutions; therefore, one must choose the desired solutions. In general, for a MOO problem, the optimal solutions obtained by individual optimization of the objectives (i.e. single-objective optimization) are not feasible solutions to the MOO problem.

There are several methods to solve the MOO problem in practice. The most common general scalar methods for multi-objective optimization is the weighted global criterion

method in which all objective functions are combined to form a single function. A weighted global criterion method is a type of utility function. One of the most general utility functions U is expressed in its simplest form as a weighted exponential sum. Two forms are used:

$$U = \sum_{i=1}^{n} w_i [f_i(\bar{x})]^p \quad f_i(\bar{x}) > 0 \quad 1 \le p < \infty \tag{5.3}$$

$$U = \sum_{i=1}^{n} [w_i f_i(\bar{x})]^p \quad f_i(\bar{x}) > 0 \quad 1 \le p < \infty \tag{5.4}$$

The most common extensions of equations (5.3) and (5.4) are (see reference [18]):

$$U = \left\{ \sum_{i=1}^{n} w_i [f_i(\bar{x}) - f_i^0]^p \right\}^{\frac{1}{p}} \quad 1 \le p < \infty \tag{5.5}$$

$$U = \left\{ \sum_{i=1}^{n} [w_i (f_i(\bar{x}) - f_i^0)]^p \right\}^{1/p} \quad 1 \le p < \infty \tag{5.6}$$

where, $\bar{x}$ is the decision vector, and $f_i^0 = \min\{f_i(\bar{x})\}$. In general, f_i^0 is usually unattainable. w is a vector of weights typically set by the decision-maker such that $\sum_{i=1}^{n} w_i = 1$ and $w_i > 0$. $\bar{x}$ is a vector of decision variables, in which the relative value of the weights reflects the relative importance of the objectives. The weights affect the final value of the objective function, the convergence speed of numerical computation and the Pareto optimal set of solutions. Equations (5.5) and (5.6) represent the summation arguments as components of a distance function which minimizes the distance between the solution point and the ideal point (also called an utopia point) in the criterion space [19]. Weighted global criterion methods are called "compromise

programming methods", as the decision-maker usually has to make a compromise between the final solution and the ideal point. If all the weights are positive, the minimum of the utility function is Pareto optimal.

A special case of a weighted global criterion method with $p=1$ is called the *weighted sum method*; its expression is the following:

$$U = \sum_{i=1}^{n} w_i[f_i(\bar{x})] \qquad f_i(\bar{x}) > 0 \tag{5.7}$$

5.3 The optimization problem of the operator's work

According to the discussions in the previous section, the optimization problem of the operator's work (in following to be called our optimization problem) is mathematically described as follows:

Objective function (in discrete form)

$$\min_{x\in C} F(\bar{x}) = \min_{x\in C} \sum_{i=1}^{n} w_i f_i(\bar{x}) = \min_{x\in C} \sum_{i=1}^{n}\sum_{k=1}^{m} w_i [x_{i,measured\ (t_0+k\Delta t)} - x_{i,simulated\ (t_0+k\Delta t)}]^2 \tag{5.8}$$

subject to the constraints:

$$C = \left\{ h^-_{i,(t_0+k\Delta t)} \le x_{i,simulated,(t_0+k\Delta t)} \le h^+_{i,(t_0+k\Delta t)} \quad i=1,2,...n, \quad \dot{\bar{x}} = A\bar{x} + Bu \right\} \tag{5.8a}$$

where w_i are the weights, t_0 is the starting time, and $\bar{x}$ is the objective function vector.

$\Delta t = (t_{end} - t_0)/m$, Δt is a discrete interval of time, and m is an integer which is selected by the decision-maker.

$[h^-_{i,(t_0+k\Delta t)}, h^+_{i,(t_0+k\Delta t)}]$ are the tolerances of $x_{i,simulated\ (t_0+k\Delta t)}$ at time $t = (t_0 + k\Delta t)$.

$\dot{\bar{x}} = A\bar{x} + B\bar{u}$ is the simulation model of the helicopter dynamics.

$$f_i(x_{i,(t_0+k\Delta t)}) = [x_{i,measured\ (t_0+k\Delta t)} - x_{i,simulated\ (t_0+k\Delta t)}]^2 \tag{5.9}$$

$\bar{x} = [x_1, x_2, ...x_n]^T$, $x_i = x_i(\bar{y},\ t)$ $i = 1,2,...,n$.

$\bar{y} = [y_1, y_2, ...y_j]$, where $\bar{y}$ is a vector of decision variables, and j is the number of decision variables. Some components of $\bar{y}$ need to be optimized while others are constants. The components which need to be optimized depend on the combination of trim and subtrim functions selected by the POM operator.

Equation (5.9) gives the square of the error between measured and simulated values at time $t = (t_0 + k\Delta t)$. $x_{i,measured\ (t_0+k\Delta t)}$ are the measured data and are constant. $x_{i,simulated\ (t_0+k\Delta t)}$ are determined by the helicopter simulation model. The initial condition changes will affect the helicopter dynamics output and the objective function evaluation.

The objective function is here expressed in a discrete form because a POM operator can only obtain the numerical results of measurements and simulations. Practically, for helicopter dynamic problems, it is impossible to obtain the expression of analytical solutions unless approximation techniques such as the numerical fitting techniques are used. However, the validity range of such approximations is likely to be very small.

5.4 Objective function weights

Our optimization objective is to find the minimal differences between the measured and simulated response by changing the simulation model initial conditions. The

differences between the measured data and the simulated results can be expressed by distances or by areas. Equation (5.8) is a combination of Least Square functions. For this type of function, mathematically, the weight vector, set by the decision-maker, affects the final value of the objective function and the convergence speed, but does not influence the final decision variable $\bar{x}$.

If we take the example, $\min_{x \in C} F(\bar{x}) = w_1(a_1 - x_1)^2 + w_2(a_2 - x_1)^2$, where $w_1 > 0$, $w_2 > 0$, a_1 and a_2 are constants, we see that the objective function consists of two sub-objective functions and no matter how the weights w_1, w_2 are changed, the accurate decision vector (or solution) is always $\bar{x}^* = (a_1, a_2)$.

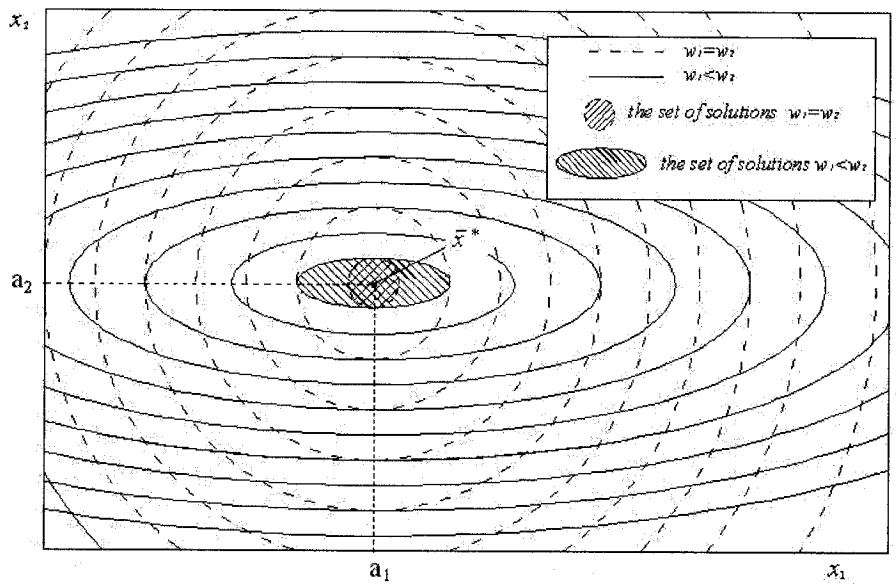


Figure 5-2 Two-dimensional top view-contours with different weights

In practice, for MOO problems, the accurate solution (also called the ideal solution) cannot usually be obtained by numeral calculation. However, approximate values

obtained are a set of solutions rather than a unique solution. Different weights may lead to different sets of solutions. Figure 5-2 shows two-dimensional top-view contours with different weights and different sets of solutions where $\bar{x}^* = (a_1, a_2)$ is the ideal solution. We see that the weights affect the sets of solutions and reflect the relative importance of the sub-objective functions. The more important the sub-objectives are, the bigger the corresponding weights. It is obvious that weights are some of the factors which affect the convergence speed in numerical computing. This is the reason why the decision-maker usually has to think of the weights carefully and make a compromise between the final solution and the ideal point. Usually we choose the product of the weights and sub-objectives to be of the same order of magnitude.

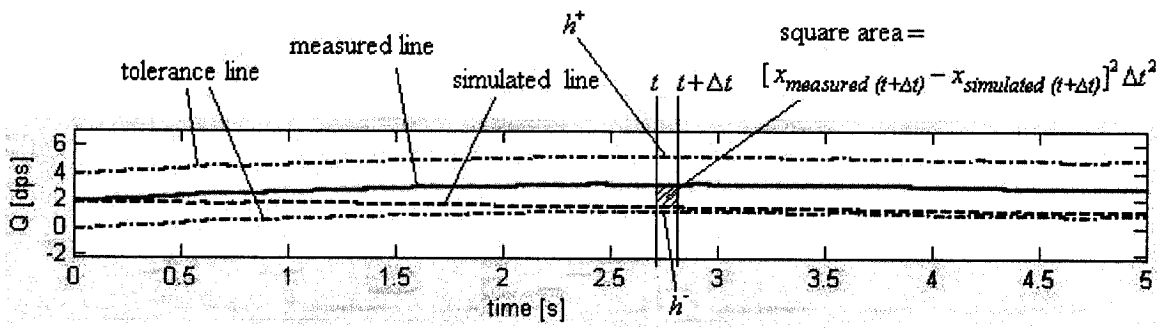


Figure 5-3 Difference between measured data and simulated result

The different terms in our objective function, excluding the weights, have different units and orders of magnitude. We therefore non-dimensionalize these terms to build up our weights. Figure 5-3 shows the difference between a measured quantity line and the simulated curve. In the graph there are four curves, two tolerance curves (red colour), simulated curve (blue colour) and measured data (green colour). In a certain time

interval $[t,t+\Delta t]$, if Δt is considered very small, we can use the measured and simulated values at $t+\Delta t$ as the averages in the interval $[t,t+\Delta t]$.The square difference of the average between measured and simulated values can be expressed as $[x_{measured\ (t+\Delta t)} - x_{simulated\ (t+\Delta t)}]^2$, so the square shaded area is $[x_{measured\ (t+\Delta t)} - x_{simulated\ (t+\Delta t)}]^2 \Delta t^2$. The smaller the area is for a given Δt, the closer the simulated value is to the measured value. If we uniformly discretize the total interval $[t_0, t_{end}]$ into 'm' small time steps Δt, the total of square area S_{m_s} between the measured values and simulated value can be expressed as follows:

$$S_{m\_s} = \sum_{k=1}^{m} \{[x_{measured\ (t_0+k\Delta t)} - x_{simulated\ (t_0+k\Delta t)}]^2 \Delta t^2\} \tag{5.10}$$

where $\Delta t = (t_{end} - t_0)/m$, and m is a positive integer.

The total of square areas between the two tolerance lines is:

$$S_{tolerance} = \sum_{k=1}^{m} \{[h^+_{(t_0+k\Delta t)} - h^-_{(t_0+k\Delta t)}]^2 \Delta t^2\} \tag{5.11}$$

From equations (5.8), (5.10), and (5.11) we obtain the following weight expression:

$$w = 1/\sum_{k=1}^{m} [h^+_{(t_0+k\Delta t)} - h^-_{(t_0+k\Delta t)}]^2 \tag{5.12}$$

Due to the fact that $h^+ - h^- = C_h$, where C_h is a constant, equation (5.12) becomes:

$$w = \frac{1}{mC_h^2} \tag{5.13}$$

where m the number of time intervals is defined in equation (5.8). For numerical convenience, we use the following weight expression:

$$w_i = \frac{100}{mC_{hi}^2} \tag{5.13a}$$

because in the general case, $\frac{1}{mC_{hi}^2}[x_{i,measured\ (t_0+k\Delta t)} - x_{i,simulated\ (t_0+k\Delta t)}]^2 < 1$ after optimization.

Our objective function is finally expressed as follows:

$$\min_{x\in C} F(x) = \min_{x\in C} \sum_{i=1}^{n} w_i f_i(\bar{x}) = \min_{x\in C} \sum_{i=1}^{n}\sum_{k=1}^{m} \frac{100}{mC_{hi}^2}[x_{i,measured\ (t_0+k\Delta t)} - x_{i,simulated\ (t_0+k\Delta t)}]^2 \tag{5.14}$$

The constraints remain as expressed in equation (5.8a).

$$C = \left\{ h^-_{i,(t_0+k\Delta t)} \le x_{i,simulated,(t_0+k\Delta t)} \le h^+_{i,(t_0+k\Delta t)} \quad i = 1,2,...n, \quad \dot{\bar{x}} = A\bar{x} + Bu \right\} \tag{5.8a}$$

Equation (5.14), (5.8a) define the multi-objective optimization problem that is to be solved. We turn to the proposed optimization algorithms in Chapter 6.

CHAPTER 6

OPTIMIZATION ALGORITHMS

In chapter 5, we have described and presented the optimization problem which is to be solved. In this chapter, we will discuss algorithms to solve optimization problems and introduce the three algorithms, Hooke and Jeeves' method (pattern search), Nelder and Mead's method, and genetic algorithm (GA), which are selected to solve the present optimization problem.

6.1 Introduction of optimization algorithms

Conventional optimization algorithms are given by indirect search methods and direct search methods. Indirect search methods ultimately involve solving an equation (or equations) rather than searching for an optimum. Usually, the objective functions and constraints are written in the form of analytical functions, inequalities or equalities. These analytical functions enable us to replace the original optimization problems by others easier to solve; frequently they depend heavily on a differential approach, and they are often used in the case when the gradient information may be found easily and the objective function is continuous and differentiable in the optimization domain.

Direct search methods are used for solving optimization problems which do not require any information about the objective function gradient. They only require the objective functions to be computable either from a formula or a series of formulas; even

a table of values can be enough. The objective functions are not required to have smoothness or even continuity. They are only required to be unimodal functions in the optimization domain. Direct methods search a set of points around the current point, looking for one point at which the value of the objective function is lower than the value at the current point. These methods proceed by an arbitrary point (or some arbitrary points) used as a starting point (or starting points), step by step toward the optimization point by successive improvements.

Since John Holland proposed the genetic algorithm (GA) in the 1960's [17], genetic algorithms (GAs) have been widely used in a variety of complex problems. One of the fields is optimization. These algorithms also do not require derivative information. GAs can solve problems which conventional direct methods cannot solve such as multi-extremum problems.

In this thesis, we select three methods to solve our optimization problem: two direct methods and a GA approach. The goal of starting off with three different methods is to find a fast method to optimize the decision vector.

6.1.1 Convexity and unimodality

In optimization problems for direct search methods, we often refer to convexity (or convex) and unimodality (or unimodal), because of the fact that these methods play significant roles in the extremum theory.

A function $f(x)$ in the n-dimensional real Euclidean space R^n is said to be convex [20] if:

$$f[\lambda x_1 + (1-\lambda)x_2] \le \lambda f(x_1) + (1-\lambda) f(x_2) \tag{6.1}$$

for any $x_1, x_2 \in R^n$ and all λ, $0 \le \lambda \le 1$. Figure 6-1 illustrates the concept of a convex function. It is obvious that if functions $f_i(x)$ are convex with $i = 1,2...n$ then $f(x) = \sum_{i=1}^{n} w_i f_i(x)$ $w_i > 0$ is also convex.

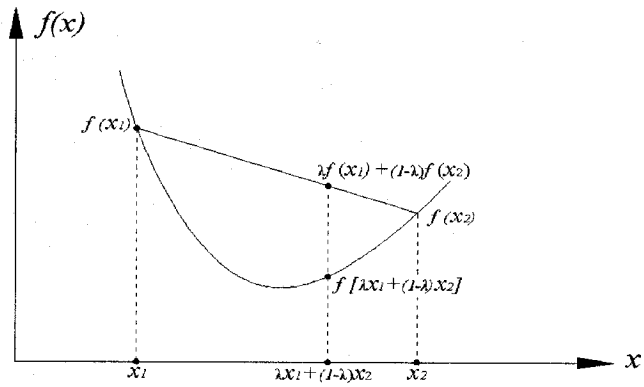


Figure 6-1 A convex function

Two important special cases of convex functions are "strictly" and 'strongly" convex. The function $f(x)$ on R^n is called strictly convex [21] if for any $x_1 \ne x_2$, $x_1, x_2 \in R^n$ and all λ, $0 < \lambda < 1$

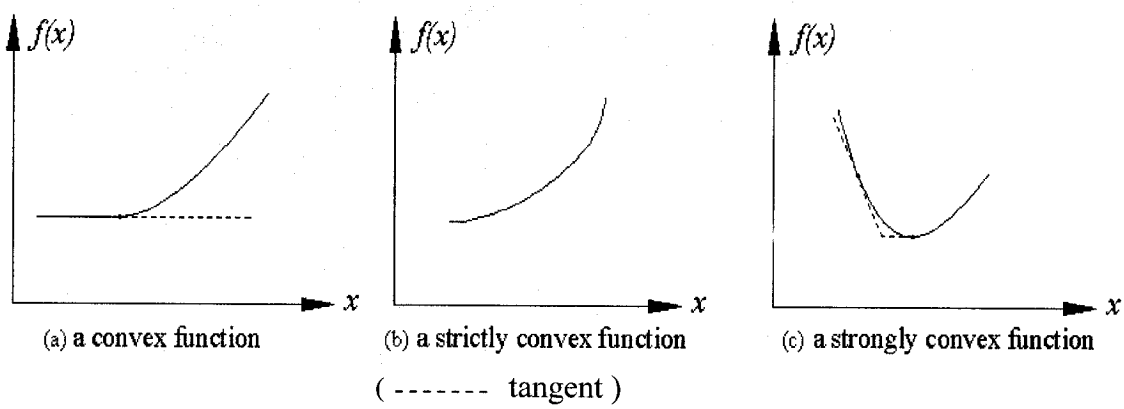
$$f[\lambda x_1 + (1-\lambda)x_2] < \lambda f(x_1) + (1-\lambda) f(x_2) \tag{6.2}$$

The function is called strongly convex with a constant $l > 0$ if for any $x_1 \ne x_2$, $x_1, x_2 \in R^n$ and all λ, $0 \le \lambda \le 1$

$$f[\lambda x_1 + (1-\lambda)x_2] \le \lambda f(x_1) + (1-\lambda) f(x_2) - l\lambda(1-\lambda)\|x_1 - x_2\|^2 / 2 \tag{6.3}$$

Clearly, a strongly convex function is strictly convex. Figure 6-2 shows the convexity types.

We can prove that the function $f(x)=\sum_{i=1}^{n} w_i(a_i-x_i)^2$ is also a strongly convex function if the function $f_i(x)=w_i(a_i-x_i)^2$ ($w_i>0, a_i=\text{constant}$) is a strongly convex function. A strictly convex function is monotone [20].



(a) a convex function (b) a strictly convex function (c) a strongly convex function

(------- tangent)

Figure 6-2 Types of convex function

Another important characteristic of a convex function which we often use in optimization problems is unimodality. A function is unimodal if there is a path from every point x to the optimal point along which the function continually increases or decreases. A strictly convex function has the characteristics that the straight line path from every point x to the optimal point is strictly increasing or decreasing and one local optimum corresponds to the global optimum. Equation (5.9) is a strongly convex function, so the equation (5.14) is also a strongly convex function which can be solved by use of the direct method.

Indirect search methods cannot be applied to our optimization problem. The reasons are: (1) the simulated values and the flight test are the tables of values rather than functional expressions, (2) the decision vector differs from the objective function vector. The two vectors are linked by a black box which represents the helicopter dynamics, and we cannot find the expressions for the first partial derivatives of the objective function with respect to the decision vector.

6.1.2 Direct search methods and GAs

In the previous section, we described direct search methods and indirect search methods. Our optimization problem can be solved by use of direct search methods or GAs. Direct search methods are effective when Newton-like methods are inappropriate or inapplicable. They are effective for a nonlinear function minimization when first partial derivatives of $f(x)$ cannot be calculated.

Direct search methods are divided into two major categories [22]: (1) ***elimination*** techniques, which narrow down the region containing the optimum until this region is acceptably small, and (2) ***climbing*** procedures which cautiously move in directions where, based on local measurement, the objective appears to improve. Any direct optimization method using past information to generate better points is called a climbing procedure. From another point of view, the problem is reduced to reaching a specified minimum acceptable level of performance in as few trials as possible.

Direct search methods include Multivariate Grid search, Hooke and Jeeves' method, and Nelder and Mead's method. Multivariate Grid search starts at a point, evaluates the objective function at the $3n-1$ surrounding points, where n is the dimensionality of decision vector. If we have a six-dimensional problem, this problem would amount to $3^6-1=728$ function evaluations at each iteration, so the greatest difficulty with the method is its inefficiency.

Hooke and Jeeves' method starts at one point and is more effective than Multivariate Grid search. This method displays the work procedure of the POM operator and is used to optimize our problem. It will be introduced in section 6.2.

Nelder and Mead's method is different from multivariate Grid search and pattern search methods. It starts calculation with $(n+1)$ points and uses the new better value to replace the worst one as pattern search method does. This method is explained in section 6.3.

GAs differ from conventional search methods [23]. Between GAs and conventional search methods, the most significant differences are:

(1) GAs search a population of points in parallel, and not a single point.

(2) GAs use probability to select population, i.e. probabilistic transition rules, not deterministic ones.

(3) GAs work on an encoding of the parameter set rather than on the parameter set itself (except where real-valued individuals are used).

(4) GAs as well as the direct methods do not require derivative information or other auxiliary knowledge; only the objective function and corresponding fitness levels influence the search direction.

(5) GAs do not require objective functions to be unimodal in the optimization domain.

GA will be introduced in section 6.4

6.2 Hooke and Jeeves' method

This method, also called pattern search in certain books, dates back to 1961. This method allows one handle constraints more easily. The search consists of a sequence of ***exploratory moves*** about a base point which if successful are followed by ***pattern moves***.

For convenience, we write the objective function $\min f(\bar{x}) = f(x_1, x_2, ..., x_n)$. The procedure is the following:

(1) Choice of an initial base point $\bar{x}_1 = (x_{11}, x_{21}, ..., x_{n1})^T$ and of a step-length vector $\bar{h} = (h_1, h_2, ..., h_n)^T$, h_j for x_j variables, $j = 1, 2...n$, where x_{j1} stands for the first value of the jth component of $\bar{x}_1$.

(2) After $f(\bar{x}_1)$ is evaluated, the method proceeds by a sequence of exploratory and pattern moves. If an exploratory move leads to a decrease in the value of $f(\bar{x})$, it

is called a success; otherwise, it is called a failure. A pattern move is not tested for success or failure.

(3) Terminate the process when the step length(s) have been reduced to a predetermined small value $\overline{h}_0 = (h_{10}, h_{20}, ..., h_{n0})^T$ which we call termination step length(s).

6.2.1 Exploratory moves

The aim of an exploratory move is to acquire information about the function $f(\overline{x})$ in the neighborhood of the current base point, and to find a descent direction. The variables are further changed one by one, by adding the step length h_j. The exploration move about the base point $\overline{x}_1 = (x_{11}, x_{21}, ..., x_{n1})^T$ can be processed as follows:

- Step 1. For $j = 1$, we change the first variable x_{11} to $\overline{x}_1 + h_1\overline{e}_1$, where $\overline{e}_1$ is a unit vector in the direction of the x_1 axis, and further evaluate $f(\overline{x}_1 + h_1\overline{e}_1)$. If $f(\overline{x}_1 + h_1\overline{e}_1) < f(\overline{x}_1)$, then we replace $\overline{x}_1$ by $\overline{x}_1 + h_1\overline{e}_1$. If it is a failure, we evaluate $f(\overline{x}_1 - h_1\overline{e}_1)$. If it is a success, we replace $\overline{x}_1$ by $\overline{x}_1 - h_1\overline{e}_1$. If it is another failure, we retain the original base point $\overline{x}_1$.
- Step 2. Set $j = 2$, change the second variable x_{12}, find $f(\overline{x}_1 + h_2\overline{e}_2)$, and then go to Step 1. When we consider all n variables, we obtain a new base point $\overline{x}_2$.

The original base point $\bar{x}_1$ and the new point $\bar{x}_2$ together establish the first pattern.

- Step 3. If $\bar{x}_2 \neq \bar{x}_1$, we make a pattern move, $\bar{x}_1$ places by pattern point $\bar{p}_1 = \bar{x}_1 + 2(\bar{x}_2 - \bar{x}_1) = 2\bar{x}_2 - \bar{x}_1$, then go to Step 1 to make an exploration.
- Step 4. If $\bar{x}_2 = \bar{x}_1$, reduce each of step lengths h_j and return to Step 1. If $h_j < h_0$, stop. Otherwise go to Step 1.

6.2.2 Pattern moves and subsequent moves

A pattern move attempts to speed up the search by using information already acquired from exploration. It is invariably followed by a sequence of exploratory moves, with the aim being to find an improved direction of search in which to make another pattern move. It seems sensible to move from $\bar{x}_2$ in the direction $(\bar{x}_2 - \bar{x}_1)$, since a move in this direction has already led to a decrease in the value of $f(\bar{x})$. The procedure for a pattern move from $\bar{x}_2$ is the following:

- Step 1. Set $k = 1$. Move from $\bar{x}_{k+1}$ to pattern point $\bar{p}_k = \bar{x}_k + 2(\bar{x}_{k+1} - \bar{x}_k)$ and continue with a new sequence of exploratory moves about $\bar{p}_k$. As a result we obtain a new point $\bar{x}_{k+2}$.
- Step 2. If $f(\bar{x}_{k+2}) < f(\bar{x}_{k+1})$, then $\bar{x}_{k+2}$ is a new base point. Set $k = k + 1$ and go to Step 1. Otherwise, abandon the pattern move from $\bar{x}_{k+1}$ and continue with

a new sequence of exploratory moves about $\overline{x}_{k+1}$.

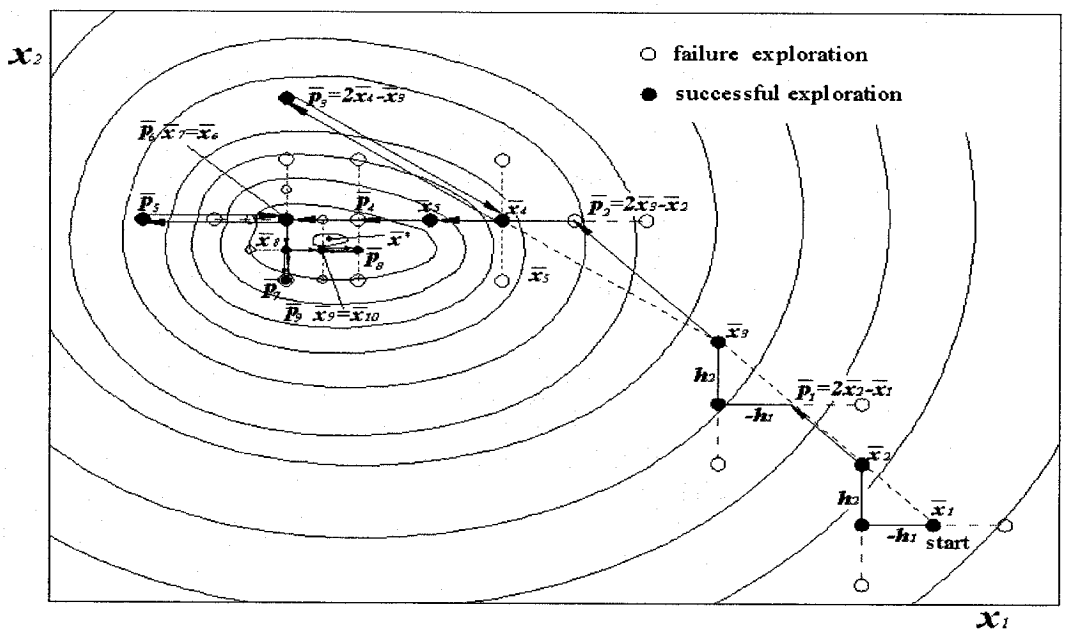


Figure 6-3 Two-dimensional pattern search (top view-contours)

Figure 6-3 shows a sequence of exploratory and pattern moves with a two-dimensional pattern search. We start from $\overline{x}_1$, and make explorations about $\overline{x}_1$, we obtain the better point $\overline{x}_2$, and the two points $\overline{x}_1$, $\overline{x}_2$ together establish the first pattern. We reason that if a similar exploration is conducted from $\overline{x}_2$, the results are likely to be better, so we double the length in the direction $\overline{x}_2 - \overline{x}_1$ and skip the local excursions to the pattern point $\overline{p}_1$, we evaluate $f(\overline{p}_1)$, which is better than $f(\overline{x}_2)$, and we continue to do explorations about $\overline{p}_1$. In the same way, we obtain pattern points $\overline{p}_2$, $\overline{p}_3$. We obtain $f(\overline{p}_3) > f(\overline{x}_4)$, we return to $\overline{x}_4$ and begin new exploration..., when we come back to $\overline{x}_7$ from $\overline{p}_5$, none of the points about $\overline{x}_7$ is better than $\overline{x}_7$, and at this time, we

reduce our step length and continue exploratory moves and pattern moves until the step length is small. We obtain the optimum $\bar{x}^*$ and $f(\bar{x}^*)$.

A flow chart representation of pattern search method is showed in Figure 6-4.

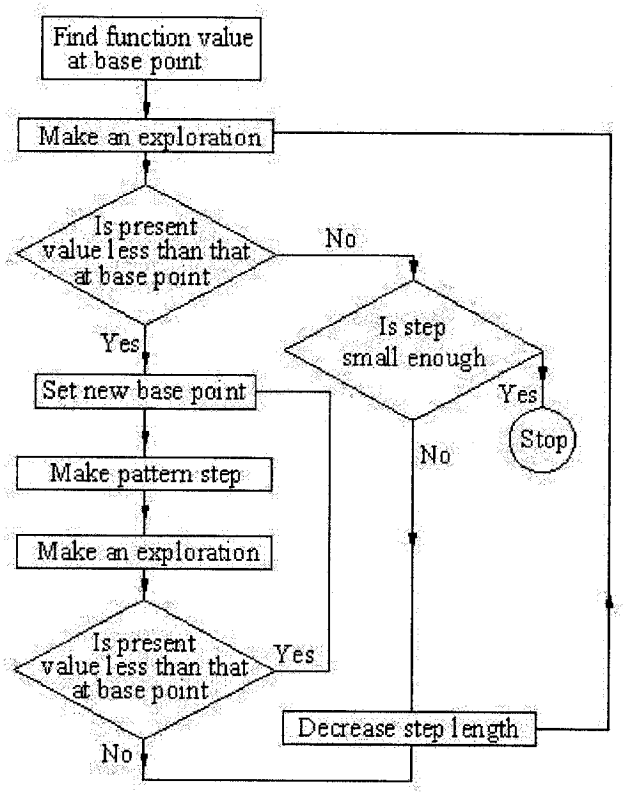


Figure 6-4 Flow chart for Hooke and Jeeves' method

6.3 Nelder and Mead's method

Nelder and Mead's method is different from pattern search which starts from one point. Nelder and Mead's method starts with $(n+1)$ points in n-dimensional space. A

set of $(n+1)$ mutually equidistant points in n-dimensional space is known as regular simplex. In two dimensions, the simplex is an equilateral triangle, and in three dimensions, it is a regular tetrahedron. The idea of the method is to compare the values of the function at the $(n+1)$ vertices of the simplex and to move the simplex towards the optimum point during the iterative process. The original simplex method maintains a regular simplex at each stage. Nelder and Mead proposed several modifications to the method which allow the simplexes to become non-regular.

The movement of the simplex in this method is achieved by the application of three basic operations: reflection, expansion, and contraction. The thinking underlying these operations becomes clear as we consider the steps in the procedure.

- Step 1. We start with $(n+1)$ points $\bar{x}_1$, $\bar{x}_2$, ..., $\bar{x}_n$ and we find

  $$f_1 = f(\bar{x}_1),\ f_2 = f(\bar{x}_2),\ \ldots,\ f_{n+1} = f(\bar{x}_{n+1}).$$

- Step 2. We find the highest function value f_h, the next highest function value f_g, the lowest function value f_l and the corresponding points $\bar{x}_h, \bar{x}_g$ and $\bar{x}_l$.

- Step 3. We find the centroid of all the points except $\bar{x}_h$. Let this be $\bar{x}_0 = \frac{1}{n}\sum_{i,i\neq h}^{n+1} \bar{x}_i$ and evaluate $f(\bar{x}_0) = f_0$.

- Step 4. It would seem reasonable to try to move away from $\bar{x}_h$. We reflect $\bar{x}_h$ in $\bar{x}_0$ to find $\bar{x}_r$ and find $f(\bar{x}_r) = f_r$. Reflection is illustrated in Figure 6-5(a). α $(\alpha > 0)$ is the reflection coefficient. We obtain $\bar{x}_r$ such that

  $$\bar{x}_r - \bar{x}_0 = \alpha(\bar{x}_0 - \bar{x}_h)$$

i.e. $\bar{x}_r = (1+\alpha)\bar{x}_0 - \alpha\bar{x}_h$ (6.4)

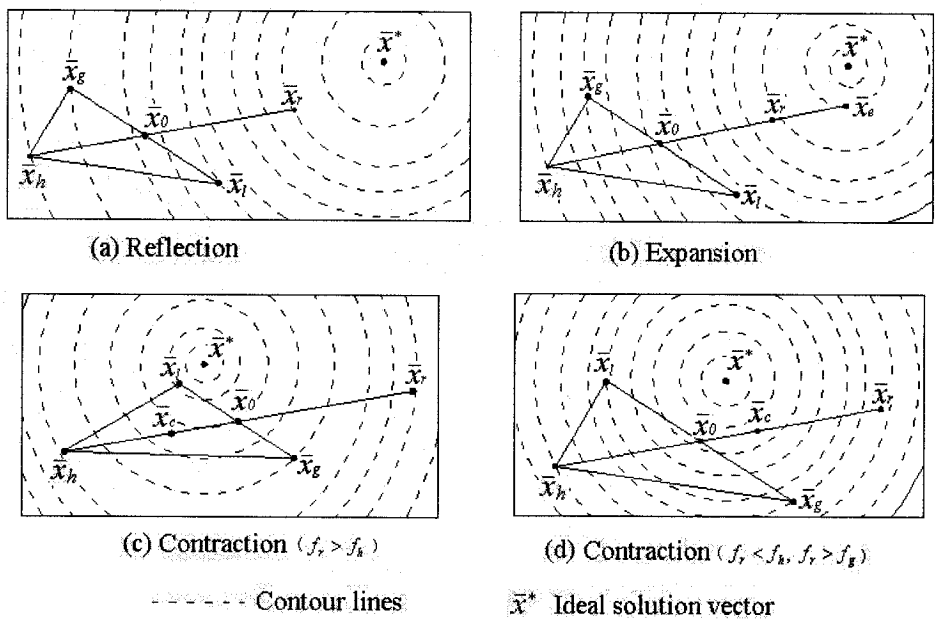


Figure 6-5 Nelder and Mead's method

• Step 5. We compare f_r with f_l.

(1) If $f_r < f_l$, we have obtained the lowest function value yet. The direction from $\bar{x}_0$ to $\bar{x}_r$ appears to be a good one to move along. We therefore make an expansion in this direction to find $\bar{x}_e$ and evaluate $f_e = f(\bar{x}_e)$. Figure 6-5 (b) illustrates the operation of expanding the simplex. With an expansion coefficient γ ($\gamma > 1$), we obtain

$$\bar{x}_e - \bar{x}_0 = \gamma(\bar{x}_r - \bar{x}_0)$$

i.e. $\bar{x}_e = \gamma\bar{x}_r + (1-\gamma)\bar{x}_0$ (6.5)

(i) If $f_e < f_l$, we replace $\bar{x}_h$ by $\bar{x}_e$ and test the $(n+1)$ points of the simplex for convergence to the minimum (see step 9). If we converge, we stop; if not, we return to step 2.

(ii) If $f_e \geq f_l$, $\bar{x}_e$ is abandoned. We have evidently moved too far in the direction $\bar{x}_0$ to $\bar{x}_r$. Instead we replace $\bar{x}_h$ by $\bar{x}_r$ which may give an improvement [step 5(1)], we test for convergence, if no convergence, we return to step 2.

(2) If $f_r > f_l$ but $f_r \leq f_g$, $\bar{x}_r$ is an improvement on the two worst points of the simplex, we replace $\bar{x}_h$ by $\bar{x}_r$ and test for convergence, if no convergence, we return to step 2.

(3) If $f_r > f_l$ and $f_r > f_g$, we proceed to step 6.

• Step 6. We compare f_r with f_h, where $f_r > f_l$ and $f_r > f_g$.

(1) If $f_r > f_h$, it appears that we move too far in the direction $\bar{x}_h$ to $\bar{x}_0$. We have to modify this move by finding $\bar{x}_c$ and f_c with a contraction step illustrated in Figure 6-5 (c). We directly proceed to the contraction, and we find $\bar{x}_c$ from:

$$\bar{x}_c - \bar{x}_0 = \beta(\bar{x}_h - \bar{x}_0)$$

$$\text{i.e. } \bar{x}_c = \beta\bar{x}_h + (1-\beta)\bar{x}_0 \qquad (6.6)$$

where β $(0 < \beta < 1)$ is the contraction coefficient.

(2) If $f_r < f_h$, we replace $\overline{x}_h$ by $\overline{x}_r$ and f_h by f_r, then we contract. Thus we find $\overline{x}_c$ from:

$$\overline{x}_c - \overline{x}_0 = \beta(\overline{x}_r - \overline{x}_0)$$

$$\text{i.e. } \overline{x}_c = \beta\overline{x}_r + (1-\beta)\overline{x}_0 \qquad (6.7)$$

This contraction illustrated in Figure 6-5 (d).

- Step 7. We compare f_c with f_h.

  (1) If $f_c < f_h$, we replace $\overline{x}_h$ by $\overline{x}_c$ and check for convergence, and if no convergence, we return to step 2

  (2) If $f_c > f_h$, all our efforts to find a value less than f_h failed, we need to proceed to the next step.

- Step 8. At this step, we reduce the size of the simplex by halving the distance of each point of the simplex from $\overline{x}_l$ which is the point to generate the lowest function value. We replace $\overline{x}_i$ by $\overline{x}_i = \overline{x}_l + 0.5(\overline{x}_i - \overline{x}_l)$, i.e. $\overline{x}_i = 0.5(\overline{x}_i + \overline{x}_l)$, then we calculate f_i for $i = 1, 2, ..., (n+1)$, test for convergence, if no convergence, we return to step 2.

- Step 9. The convergence test is based on the standard derivation of the $(n+1)$ function values which should be smaller than a pre-determined small value ε. We calculate

$$\sigma^2 = \frac{1}{(n+1)} \sum_{i=1}^{n+1} (f_i - f_{average})^2 \qquad (6.8)$$

$$\text{where } f_{average} = \frac{1}{(n+1)} \sum_{i=1}^{n+1} f_i$$

If $\sigma < \varepsilon$, all function values are very close to each other and also the points near the minimum $\bar{x}_l$. This convergence criterion is reasonable [29].

Figure 6-6 gives a flow chart representation of the Nelder and Mead's method.

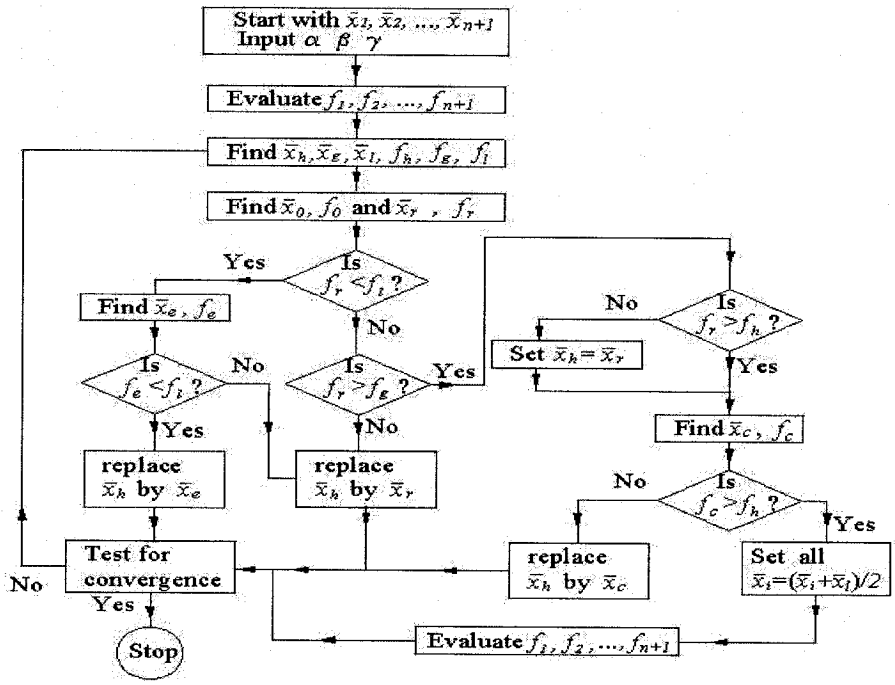


Figure 6-6 Flow chart of Nelder and Mead's method

6.4 Genetic algorithm (GA)

6.4.1 GA introduction

GA is a stochastic global search algorithm based on the natural genetics mechanics. It imitates nature with their Darwinian survival-of-the-fittest approach and the metaphor of

natural biological evolution. It differs from other search techniques, and the algorithm starts with an initial set of random solutions called a population. Generally, the structure of genetic algorithms is shown in Figure 6-7. GA works on the solution space and the coding space alternatively, evaluation and selection work on the solution space, while genetic operations work on the coding space (chromosomes). The convergence speed by use of GA is mainly affected by the mutation rate, but GA can always obtain the optimum solution as long as the mutation rate is not equal to zero. The GA operations include the major parts described below.

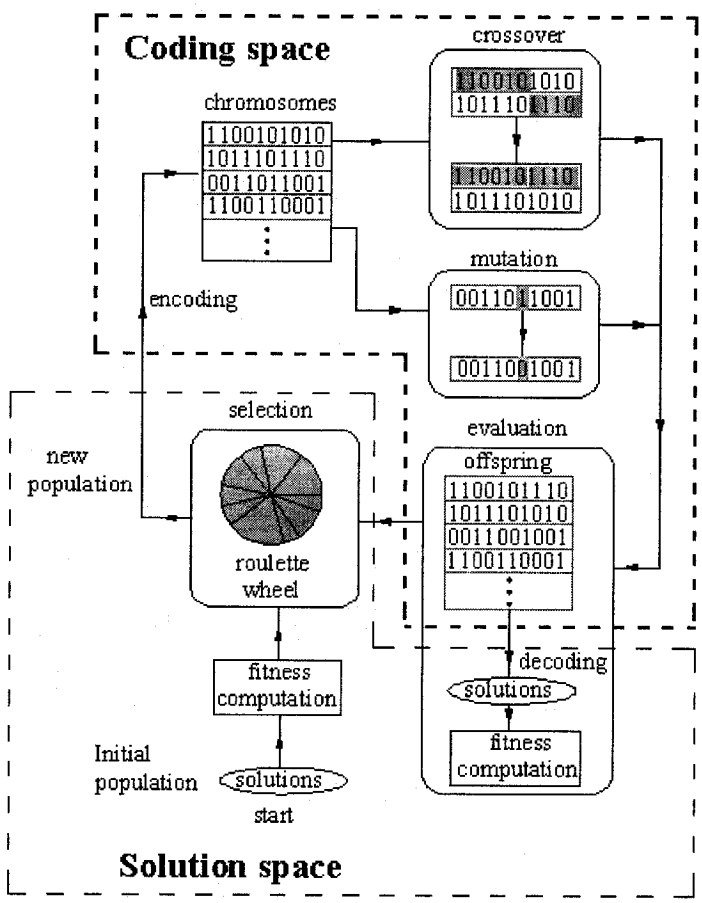


Figure 6-7 The general structure of genetic algorithms

6.4.2 Initial population

The Initial population is an initial set of random solutions in the optimization domain. Each individual in the population is called chromosome which is composed of genes and represents a solution to the problem. A chromosome can be expressed by real-valued or by a string of symbols made up by zeros and ones called genes. It is evaluated by a fitness function. We usually need to encode decision variables into binary strings and concatenate those head-to-tail to form a long string. The length of the strings is determined by the required precision, i.e. the number of zeros after the decimal point. For a variable $a_i \le x_i \le b_i$, the precision means that the domain range of the variable should be divided into at least $(b_i - a_i) \times 10^{precision}$ size ranges. The required bits (denoted by m_i) are obtained by the following expression:

$$2^{m_i - 1} < (b_i - a_i) \times 10^{precision} \le 2^{m_i} - 1 \quad . \qquad (6.9)$$

A binary string is converted into a decimal number by use of the following expression:

$$x_i = a_i + x_{decimal(substring_i)} \times (b_i - a_i)/(2^{m_i} - 1) \qquad (6.10)$$

where $x_{decimal(substring_i)}$ represents the decimal value of $substring_i$ for decision variable.

For example, a chromosome C is represented by the following string:

$$C = [b_m b_{m-1} \ldots b_2 b_1 b_0 a_n a_{n-1} \ldots a_2 a_1 a_0]$$

where b and a are made up by 0 or 1, the first value associated with the substring composed by the "b" elements and is written as follows:

$$x_{decimal(substring_i)} = \sum_{i=0}^{m-1} b_i \times 2^i \tag{6.11}$$

6.4.3 Fitness function

The fitness function actually depends on the objective function. The fitness values must be non-negative and they decide the survival or the extinction of chromosomes in the next generation. For optimization problems such as minima, the objective function must be transferred into a relative fitness function as follows:

$$F(\overline{x}) = g(f(\overline{x}))$$

where $f(\overline{x})$ is the objective function, $g(\cdot)$ transforms the objective value into a non-negative number and F is the relative fitness.

6.4.4 Selection/Reproduction

Each chromosome is evaluated by the given fitness function to decide its survival or extinction in the next generation. The better the fitness value is, the higher the probability is for a chromosome to survive. Chromosomes which have better fitness values than the average have a chance to duplicate more than once in the next generation. As a result, better chromosomes will dominate the searching space for good solutions. In most practice, a roulette wheel method is employed as the selection procedure, and by this method, a new population is selected with respect to the

probability distribution based on fitness value. The roulette wheel can be described as follows [28].

(a) Calculate the fitness value by use of the fitness function:

$$F(\overline{x}_k) = g(f(\overline{x}_k)) \qquad k = 1, 2 \ldots popsize.$$

where *popsize* denotes the number of individuals in the population.

(b) Calculate the total population fitness:

$$F_{sum} = \sum_{k=1}^{popsize} F(\overline{x}_k) \qquad k = 1, 2 \ldots popsize.$$

(c) Calculate the selection probability p_k for each chromosome $F(\overline{x}_k)$

$$p_k = F(\overline{x}_k) / F_{sum} \qquad k = 1, 2 \ldots popsize.$$

(d) Calculate the cumulative probability q_k for each chromosome $F(\overline{x}_k)$

$$q_k = \sum_{k=1}^{popsize} p_k \qquad k = 1, 2 \ldots popsize.$$

The selection process begins by rotating the roulette wheel *popsize* times. Each time, a single chromosome is selected for a new population.

6.4.5 Crossover

Crossover in the GA is a basic operator for producing new chromosomes. Crossover rate (denoted by p_c) is defined as ratio of the number of offspring produced in each generation to the number of individuals in the population (also called the population size) [24]. Based on the pre-assigned crossover rate, a certain number of chromosome

pairs are randomly chosen to execute the crossover operation, in which one chromosome in a pair will exchange part of its sub-sequence with the other (see Figure 6-7). The crossover operation allows the chromosomes to approach the desired solutions in the long run. A higher crossover rate allows exploration of more of the solution space and reduces the chances of settling to a false optimum; but if this rate is too high, it results in a waste of computation time in exploring unpromising regions of the solution space [27]. In different references, the authors used different crossover rates such as p_c=25% in reference [24] and [25]; p_c=65% in [26]; and p_c=75% in [27]. The authors do not express why they choose these values of p_c.

6.4.6 Mutation

Mutation modifies chromosomes; the mutation operation widens the searching space for the optimal solutions [28]. The mutation rate (denoted by p_m) in GAs is a very important factor and is defined as the percentage of the total number of genes in the population. Based on the pre-assigned mutation rate, a certain number of genes in the population are randomly chosen to flip from 1 to 0 or from 0 to 1 to simulate the mutation situations occurring in biological system (see Figure 6-7). If the mutation rate is too low, many genes which would have been useful are never tried out; but if it is too high, there will be too much random perturbation: the offspring will lose their resemblance to the parents and the algorithm will lose the ability to learn from the search history [24]. According to Wu and Chow [25], the mutation rate should

be $0.5\% < p_m < 5\%$, but in reference [26], the author gives the formulation $p_m = 1/popsize$; and in reference [27], the author chooses p_m=0.05, but these authors do not explain the reason why they choose such p_m.

6.4.7 The drawback of GAs

One drawback is the GA termination. GA is a random search method, it is difficult to find the special convergence criterion [23]. A common practice is to stop the genetic algorithm following a pre-specified number of generations and then test the quality of the best individual against the problem definition. If no acceptable solution is found, the GA may be restarted or a fresh search should be initiated.

Another drawback is the speed of obtaining the acceptable solutions. Because selections of initial population and parents to crossover and mutate are random, the speed is also random. When we use the number of generation as the termination of the GA, if the number is small, the acceptable solutions sometimes can not be obtained and the GA has to restart, but if the number is large, the GA takes a long time. Therefore the time required to get an acceptable value, i.e. the convergence speed, has been a central issue of GAs for a long time.

6.5 Case studies for GA

From the previous sections, we can conclude that a genetic algorithm can be used to

solve our optimization problem. We can see that the p_c and p_m (crossover and mutation rates) are the most important factors in GA, but we do not find the answer as to what values of p_c and p_m are reasonable.

We perform the following two case experiments due to the fact that we want to know if p_m, p_c are reasonable to obtain acceptable values quickly for these cases. We apply the p_m, p_c and the program to our optimization problem. In genetic algorithms, the optimum appears sometimes in earlier generation, where the probability of appearing optimum is relevant to the crossover and the mutation rate. We choose typically two cases to study p_m and p_c.

Case 1. $\max f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2)$

$$-3.0 \le x_1 \le 12.1 \qquad 4.1 \le x_2 \le 5.8$$

This function is highly multimodal as shown in Figure 6-8.

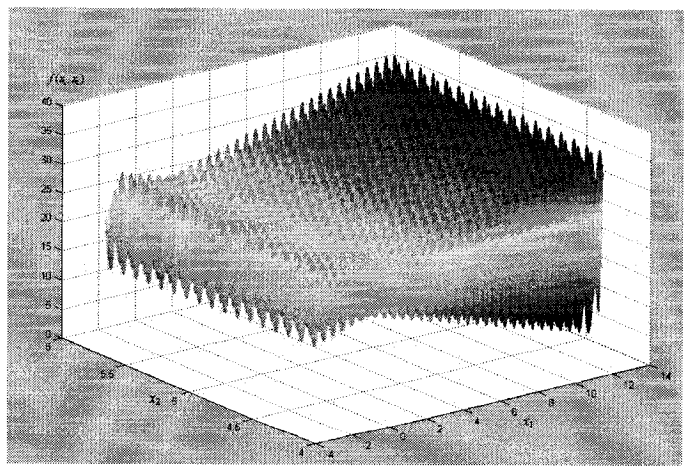


Figure 6-8 Surface of $f(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2)$

Case 2. $\min f(x_1, x_2, x_3, x_4) = 4(x_1 - 1)^2 + 3(x_2 - 2)^2 + 2(x_3 - 3)^2 + (x_4 - 4)^2$

$$-3 \le x_1 \le 4, \qquad -2 \le x_2 \le 5, \qquad -1 \le x_3 \le 6, \qquad 0 \le x_4 \le 7$$

This objective function is obviously unimodal because it is the summation of unimodal functions (see pages 76-77).

Case 1 and 2 are two extreme cases expected to encompass our optimization problems.

6.5.1 Finding the best value and decision vector

Case 1. Select the fitness function to be the same as the objective function [28], i.e.

$$F(x_1, x_2) = 21.5 + x_1 \sin(4\pi x_1) + x_2 \sin(20\pi x_2)$$

$$-3.0 \le x_1 \le 12.1 \qquad 4.1 \le x_2 \le 5.8$$

Case 2. The objective is to find the minimum, therefore we must transform the objective function into a non-negative function, and we use the fitness function as shown below:

$$F(x_1, x_2, x_3, x_4) = \frac{10}{f(x_1, x_2, x_3, x_4) + 1}$$

$$f(x_1, x_2, x_3, x_4) = 4(x_1 - 1)^2 + 3(x_2 - 2)^2 + 2(x_3 - 3)^2 + (x_4 - 4)^2$$

$$-3 \le x_1 \le 4, \quad -2 \le x_2 \le 5, \quad -1 \le x_3 \le 6, \quad 0 \le x_4 \le 7$$

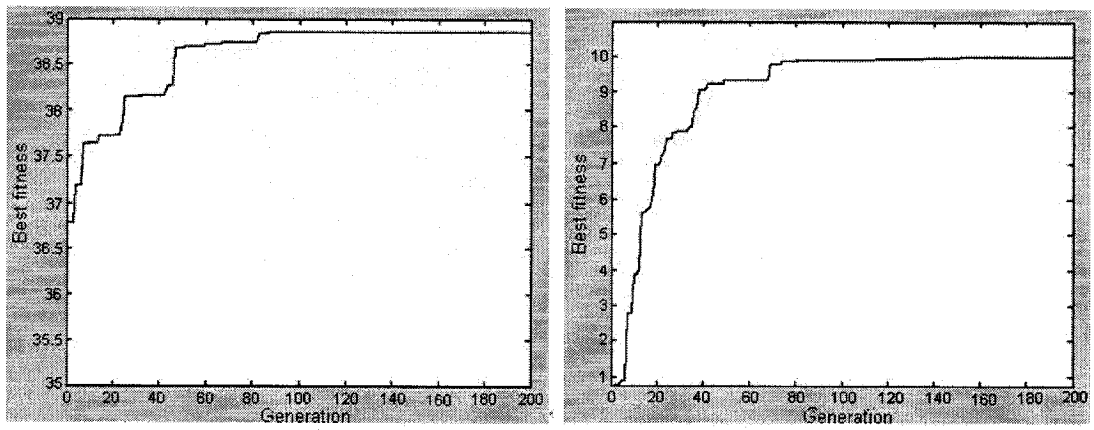
The results are the following:

Case 1. $(x_1, x_2) = (11.6255, 5.7250)$ $f_{max} = F_{max} = 38.8503$; the maximum appears in the 89th generation at this time running Matlab code. See Figure 6-9 (a).(The reference [28] results are $(x_1, x_2) = (11.6314, 5.7248)$, $f_{max} = 38.8182$)

Case 2. $(x_1, x_2, x_3, x_4) = (1.0000, 2.0000, 3.0000, 4.0003)$, $f_{min} = 9.6946 \times 10^{-8}$, $F_{max} = 10.00$;

the maximum appears in the 160th generation at this time running Matlab code. See Figure 6-9 (b).

Selecting different p_c and p_m, we can always obtain the optimization solution as long as p_m is not equal to zero and the generation is high enough, but different p_c and p_m lead to different convergence speeds.



(a) Case 1 (b) Case 2

Figure 6-9 Generation versus fitness of the cases

6.5.2. Mutation rate and acceptable values

In practice, we use the relative error to define an acceptable value, the definition of the relative error is:

$$error = \frac{\left|v_{acceptable} - v_{ideal}\right|}{\left|v_{ideal}\right|} \times 100\% \quad \text{if} \quad v_{ideal} \neq 0 \tag{6.12}$$

$$error = \left|v_{acceptable}\right| \quad \text{if} \quad v_{ideal} = 0 \tag{6.13}$$

where v_{ideal} is an ideal value and $v_{acceptable}$ is an acceptable value. We define the $error \leq 5\%$, for example, we consider that $|v_{acceptable}| \geq 95\%|v_{ideal}|$.

Following the flow chart given in Figure 6-10, we obtain the results shown in Figures 6-11 and 6-12.

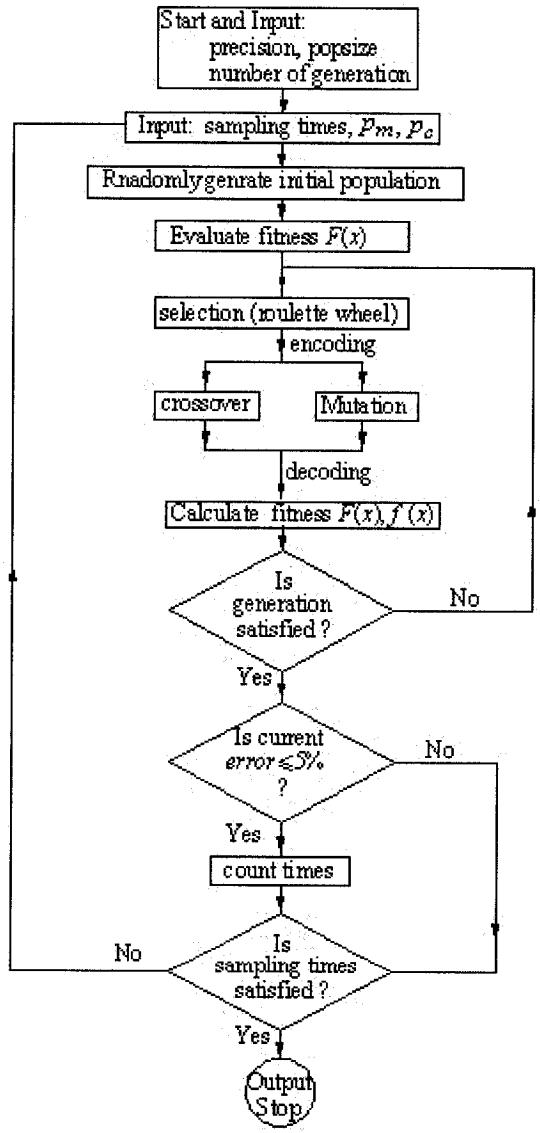


Figure 6-10 Flow chart of calculating probability

Figure 6-11 shows the relationship between the probability of $v_{acceptable}$ and the mutation rate, where $p_c = 40\%$ for case 1 and $p_c = 30\%$ for case 2. It also shows the largest probability to get the acceptable values appears where the mutation rate is about 0.08. However, if the mutation rate is too high, there will be very high random perturbation shown [24] [27]. According to Figure 6-11, the better choice of p_m should be $0.05 < p_m < 0.08$.

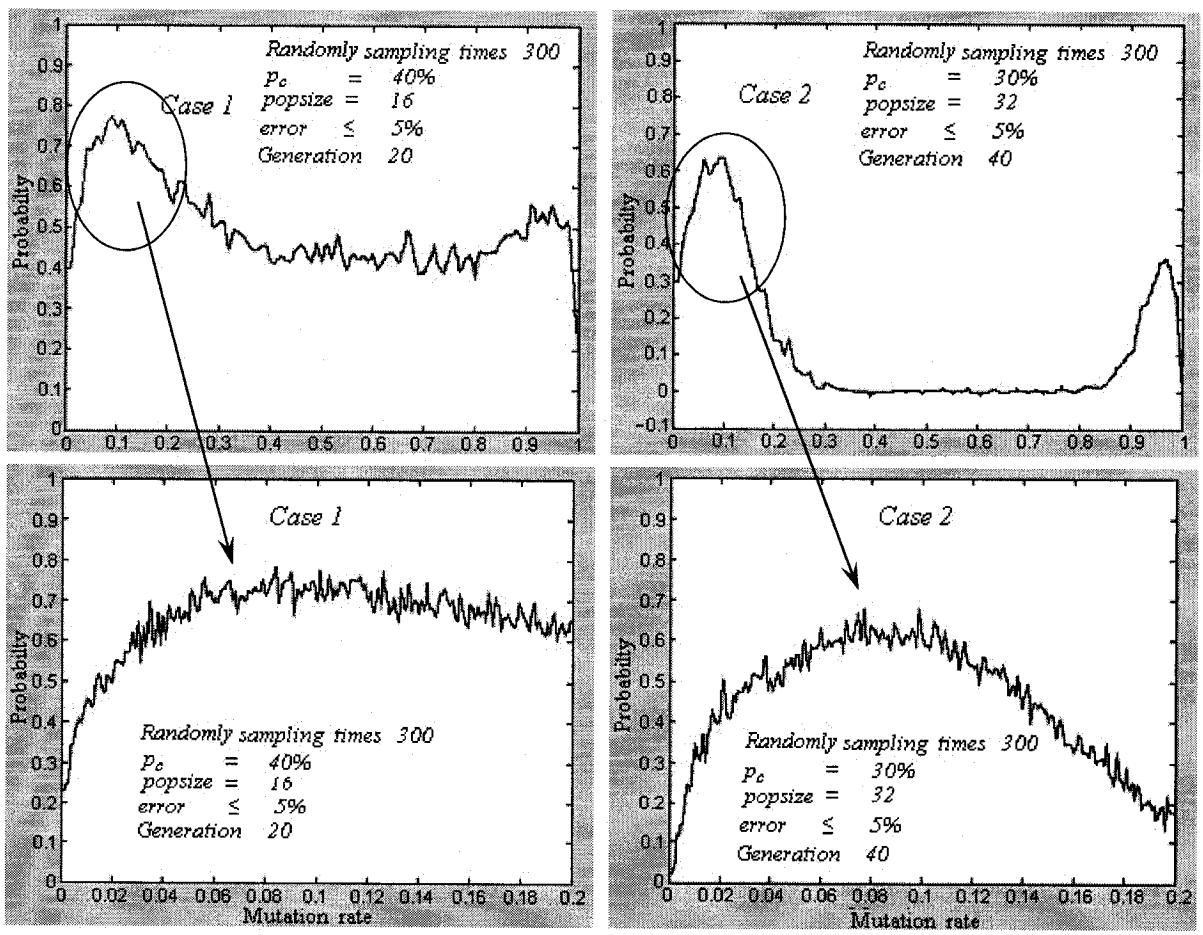


Figure 6-11 Probability of $v_{acceptable}$ versus mutation rate

Figure 6-12 shows the relationship between the probability of $v_{acceptable}$ and the

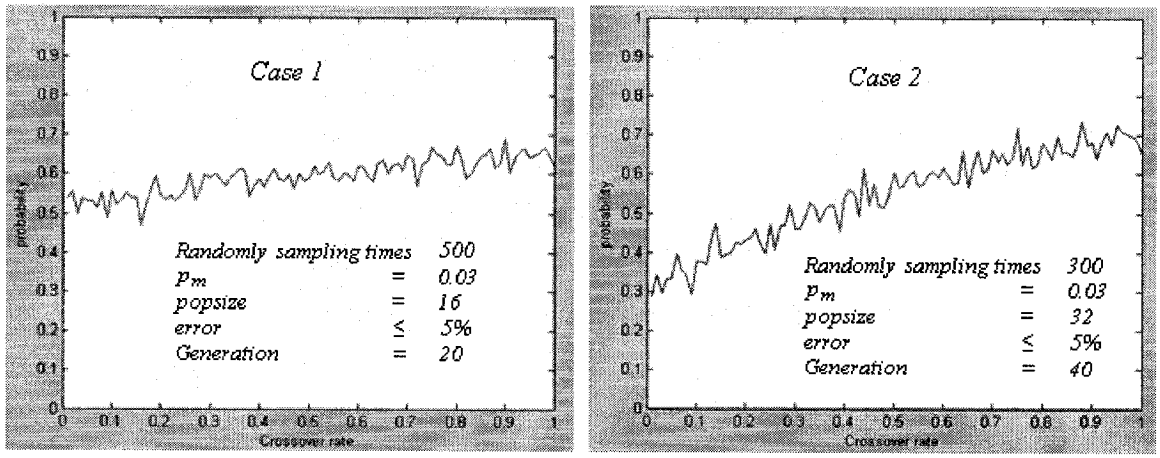


Figure 6-12 Probability of $v_{acceptable}$ versus crossover rate

crossover rate, where $p_m = 0.03$ for case 1 and case 2. It also shows the probability rises with the increase in the crossover rate. But if this rate is too high, it results in a waste of computation time in exploring unpromising regions of the solution space [27], According to [26], [27] and with Figure 6-12, the range of crossover rate can be chosen in the $0.6 < p_c < 0.8$ range.

From the two case studies and reference [26], [27], we can see the reasonable p_m and p_c should be $0.05 < p_m < 0.08$, $0.6 < p_c < 0.8$. In the later section, we use the different value of p_m to check our optimization problem.

CHAPTER 7

APPLICATION OF THE THREE ALGORITHMS

In chapter 6, we introduced in detail the three algorithms which are selected to solve our optimization problem. In this chapter, we apply the three algorithms and Matlab Graphical User Interface (GUI) techniques to the Bell 427 validation problem and develop an automatic POM software called AutoPOM. We also compare the convergence speeds of the three algorithms. The development and final implementation of an optimization theory based Automatic POM (AutoPOM) algorithm and Graphic User Interface (GUI) software is the primary original contribution of the present master's thesis research.

7.1 The software AutoPOM

7.1.1 Description of AutoPOM

In section 4.4, we presented the use of the software POM and explained the reasons behind the need to develop AutoPOM. AutoPOM is the combination of POM and the three optimization algorithms presented in chapter 6. It runs in the Matlab environment as POM does. In order to use AutoPOM easily, a Matlab Graphical User Interface (GUI) was used to create our GUI. This interface enables the AutoPOM user to operate the application without knowing the commands which would be required by a command

line interface, and to get the best validation results without having to observe the output plot changes during the validation process. For the present research, it was vitally important both to develop the theory and also to develop a product that could be used by engineers in industry.

The AutoPOM interface that was developed is shown in Figure 7-1.

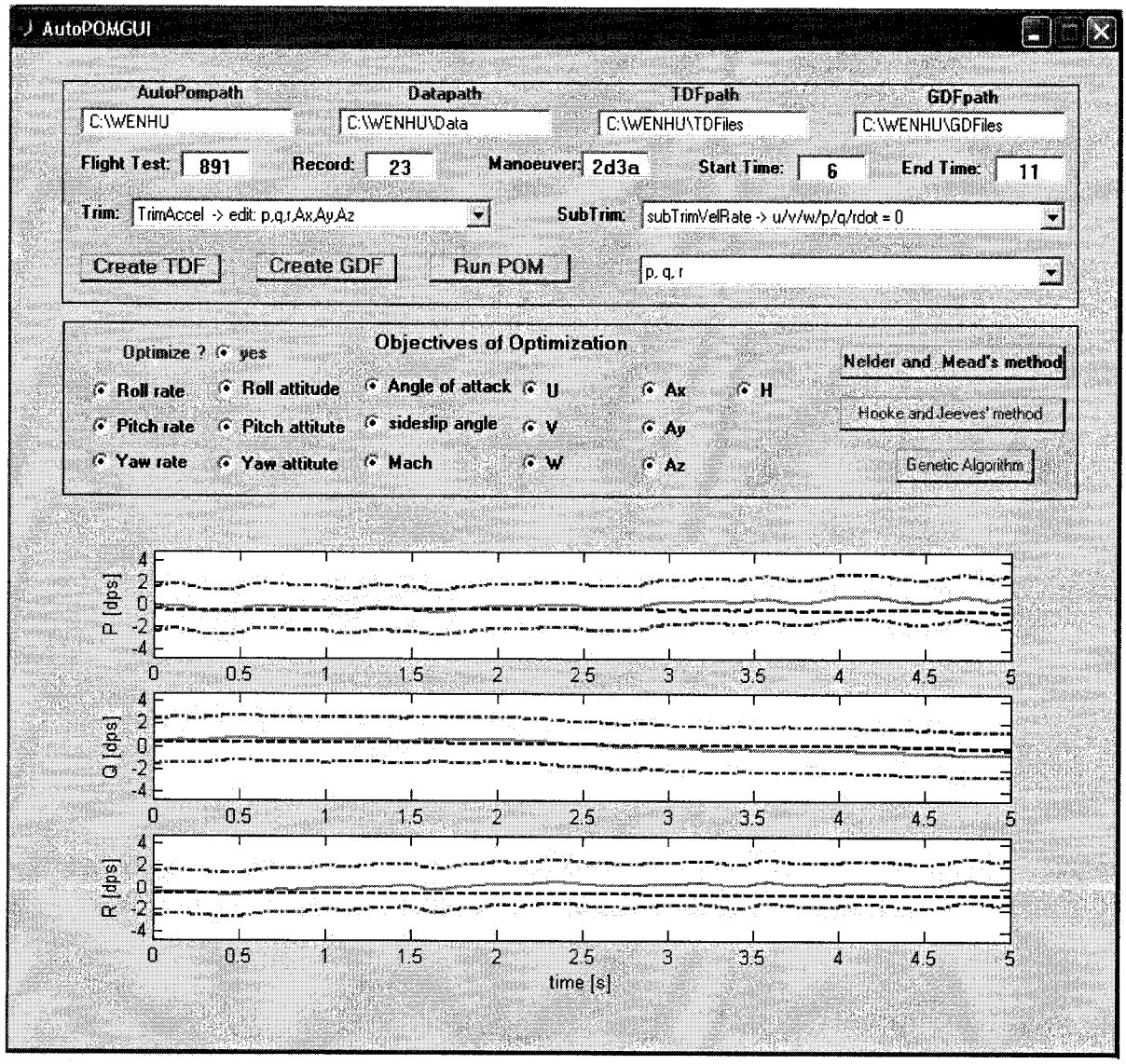


Figure 7-1 AutoPOM interface

7.1.2 The AutoPOM readme

The programming environment is Matlab version 6.5. The program is launched by typing its name, ***AutoPOMGUI***, in the Matlab command window. The boxes and Push buttons of the interface are:

1. Box **AutoPompath** indicates the path of simulation model and AutoPOMGUI.m, AutoPOMGUI.fig.
2. Box **Datapath** indicates the path of the flight test data.
3. Box **TDFpath** indicates the path in which the TDF file will be saved, in which rewriteTDF.m and writeTDFGUI.m should be included.
4. Box **GDFpath** indicates the path in which the GDF file will be saved, in which *writeGDFGUI.m* should be included. User should change the tolerances according to Table 4.2 in Chapter 4.
5. Box **Flight** test and box **Record** indicate flight case.
6. Box **Maneuver** from Table 4.1.
7. Box **Trim.** User can select a trim condition from the list.
8. Box **SubTrim.** User can select a sub-trim condition from the list.
9. Push button **Create TDF** creates the Test Definition File(TDF). The TDF is saved in the folder TDFiles which box **TDFpath** indicates.
10. Push button **Create GDF** creates the Graphical Definition File (GDF), which is saved in the folder GDFiles which box **GDFpath** indicates.

11. Push button **Run POM** starts POM software from NRC. The results are available for plotting.
12. **Objectives of optimization** allows the user to choose which parameters are selected in the objective function. User must firstly choose the radio button "yes" in order to activate the optimization objectives if he/she wants to perform the optimization.
13. Push buttons **Nelder and Mead's method, Hooke and Jeeves' method, genetic algorithm** allow the user to choose any one of these optimization algorithms. The optimization procedure starts with the current values of initial conditions from TDF. The starting parameters are those from flight test. The optimized values replace the initial values in the TDF file. The user can run again Create TDF to restore the initial values.

7.2 Application examples of the three optimization algorithms

7.2.1 Results obtained with the three optimization algorithms

Our thirteen decision variables are $p, q, r, A_x, A_y, A_z, \phi, \beta, Mach, ROC, \dot{p}, \dot{q}, \dot{r}$. The variables are different if a combination of trim and subtrim functions is differently selected (see section 4.3).

Our sixteen optimization objectives are $p, q, r, \phi, \theta, \psi, \alpha, \beta, Mach, u, v, w, a_x, a_y, a_z, H$. We can choose any objectives to optimize. The optimal solution minimizes the

differences between the simulated results and the recorded data.

In the multi-variable objective function optimization, the final decision vector is not unique but rather belongs to a set of solution vectors. For a typical optimization problem, different optimization algorithms may give different solutions. In addition, a given algorithm, such as pattern search, used with different step-length and start point, mar leads to different solutions.

Table 7.1 The initial condition parameters before and after optimization

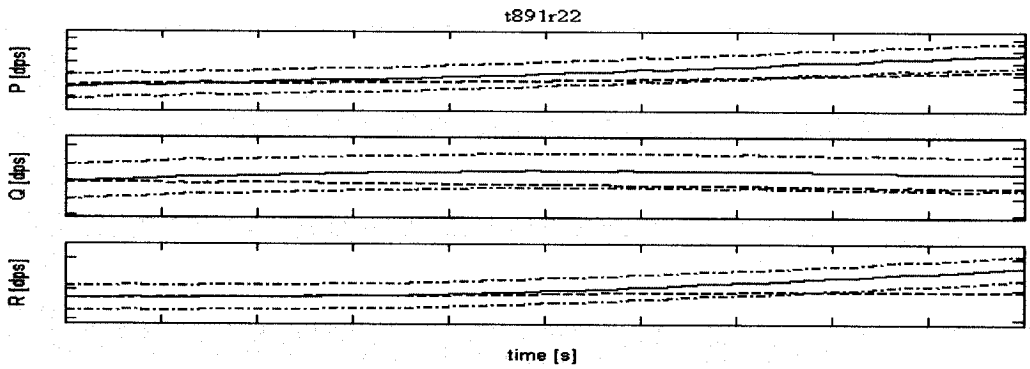
Flight case t891r22				
Manoeuver 2d3a, Start time 9 sec, End time 14 sec				
Optimization parameters	Initial conditions			
	Before optimization	Results after optimization		
		pattern search	Nelder and Mead's method	genetic algorithm
ROLL_RATE (p) (deg/s)	0.178	0.087	0.6023	0.433
PITCH_RATE (q) (deg/s)	1.82	2.320	2.3072	2.607
YAW_RATE (r) (deg/s)	-1.458	-0.858	-1.0042	-1.071
PDOC $(\dot{p})$ (deg/s^2)	0	1.5	1.9381	-2.236
QDOC $(\dot{q})$ (deg/s^2)	0	2	1.5847	1.967
RDOC $(\dot{r})$ (deg/s^2)	0	-0.5	-0.1317	0.205
Value of objective function	16.0379	1.1203	1.3124	1.3217

Figures 7-2 to 7-7 show the sample of time history plots of flight case t891r22 before and after optimization by use of the three algorithms. The two red coloured lines are the tolerance lines, the blue colour line is the simulated result, and the green coloured line is the measured data. The combination of the trim function TrimAngAccel and the subtrim function subTrimVelRate is selected. The selection allows us to edit initial conditions $p, q, r, \dot{p}, \dot{q}, \dot{r}$. Table 7.1 only presents the initial condition parameters which change before and after optimization. We can see the initial conditions are

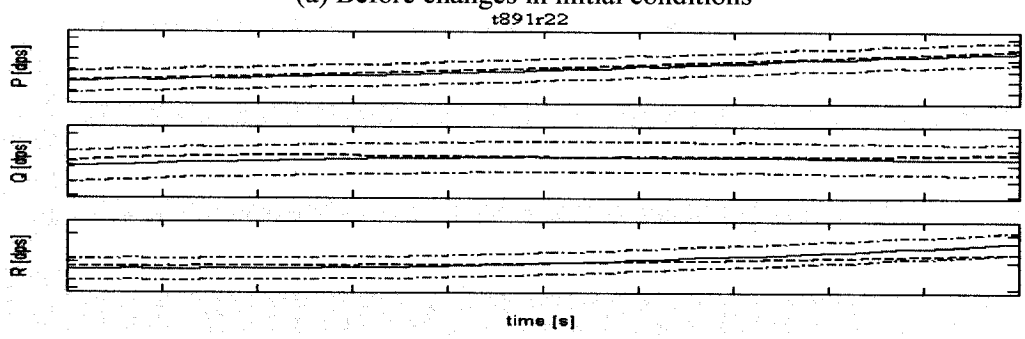
different after optimization by use of different algorithms. From Figure 7-2 to Figure 7-7, the optimization results are all in the set of the solution because the simulated results meet the level D requirements in the Advisory Circular AC 120-63 [15].

By comparison of the time history plots (see Figure 7-2 to Figure 7-7), obvious differences are found between before and after optimization results. For example, in Figure 7-2, before optimization, the simulated yaw rate (r) is off the tolerance, after optimization, the simulated yaw rate (r) is within the tolerance. Another example, in Figure 7-3, before optimization, the Euler angles ϕ, θ and ψ are all off the tolerances, after optimization, Euler angles are all within the tolerances. When we compare the results obtained with the three optimization algorithms, we find small differences between the algorithms.

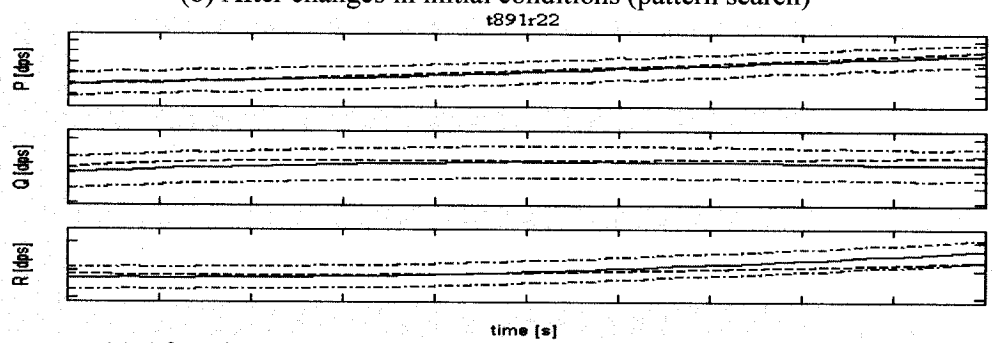
For the flight case t891r22, by use of AutoPOM (the latest POM version 7.2), the simulated results (i.e. simulated plots) obtained with the three optimization algorithms are within the tolerances required by Table 4.2. We tested the three algorithms by 4 flight cases t891r21, t891r23 to t891r25 and 11 flight cases t904r10 to t904r20, we quickly obtained validation results (i.e. the changed initial conditions) by use of AutoPOM, this validation results make 80% of the output plots (i.e. 16 output plots) within the tolerances [14]. These demonstrate that the POM version 7.2 is "true" for the flight cases.



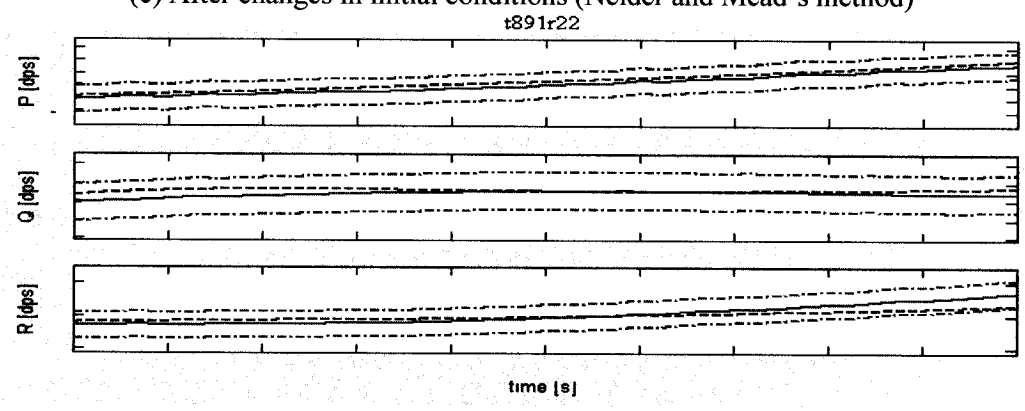
(a) Before changes in initial conditions



(b) After changes in initial conditions (pattern search)

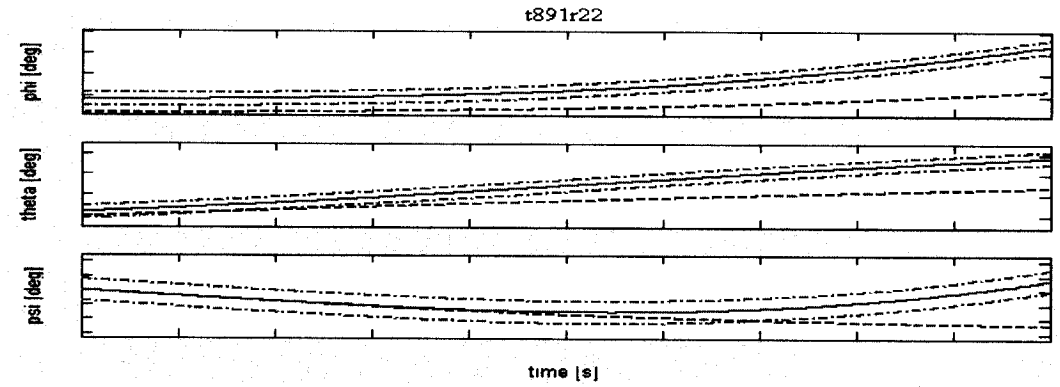


(c) After changes in initial conditions (Nelder and Mead's method)

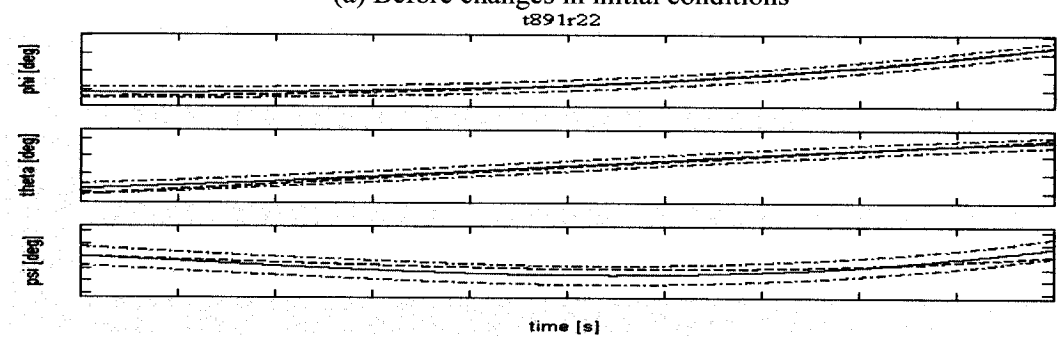


(d) After changes in initial conditions (genetic algorithm)

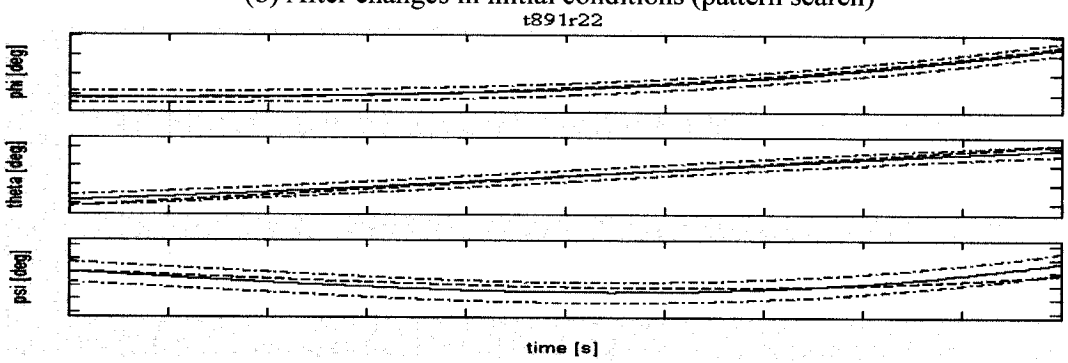
Figure 7-2 Comparison of time history plots of angular velocities p, q, r



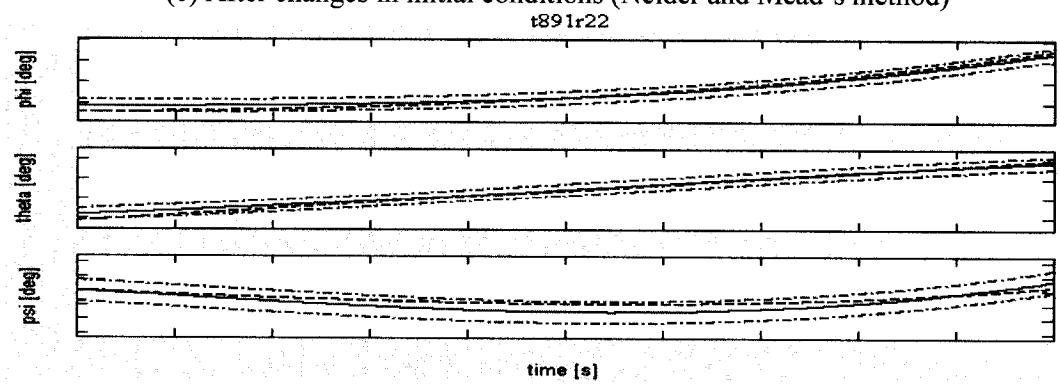
(a) Before changes in initial conditions



(b) After changes in initial conditions (pattern search)

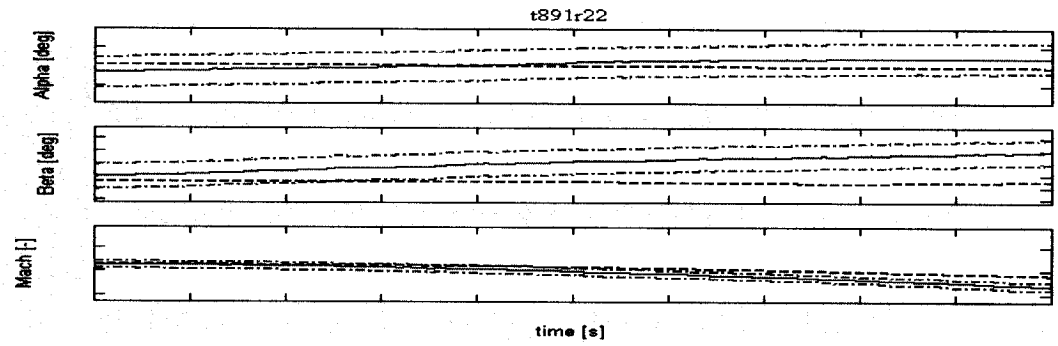


(c) After changes in initial conditions (Nelder and Mead's method)

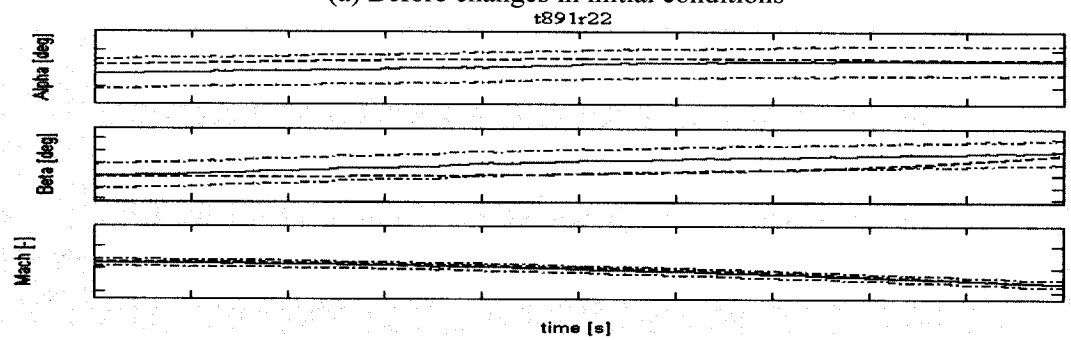


(d) After changes in initial conditions (genetic algorithm)

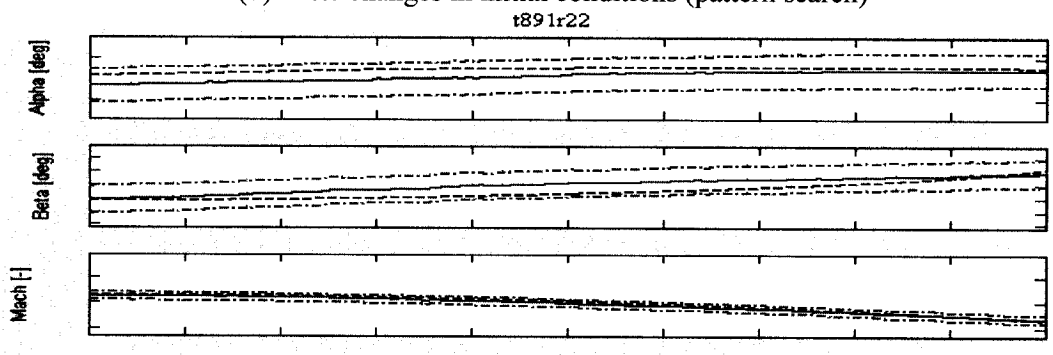
Figure 7-3 Comparison of time history plots of Euler angles ϕ, θ, ψ



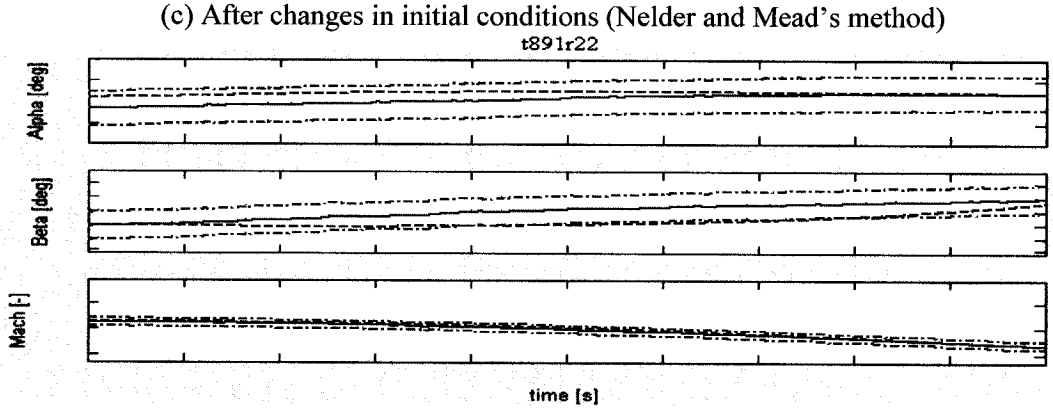
(a) Before changes in initial conditions



(b) After changes in initial conditions (pattern search)

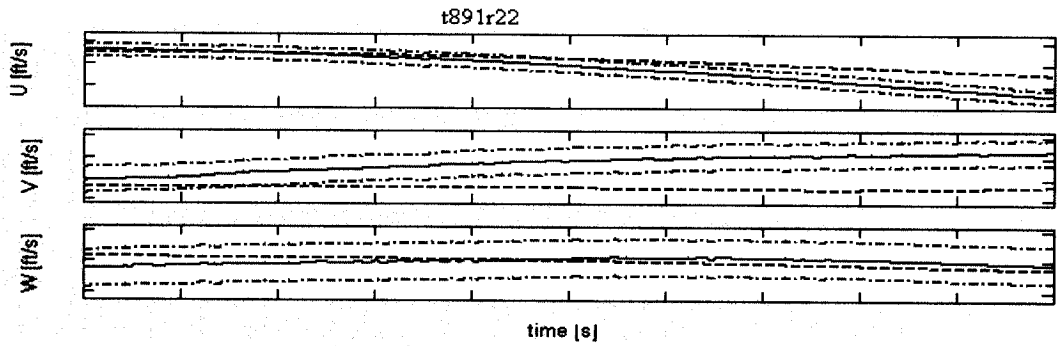


(c) After changes in initial conditions (Nelder and Mead's method)

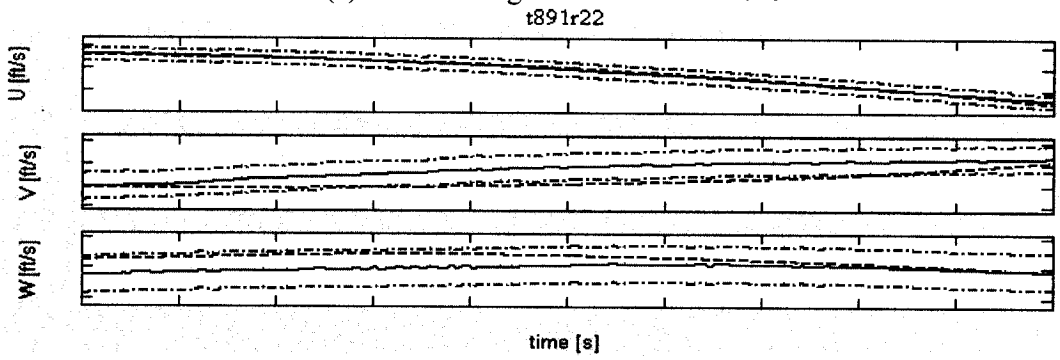


(d) After changes in initial conditions (genetic algorithm)

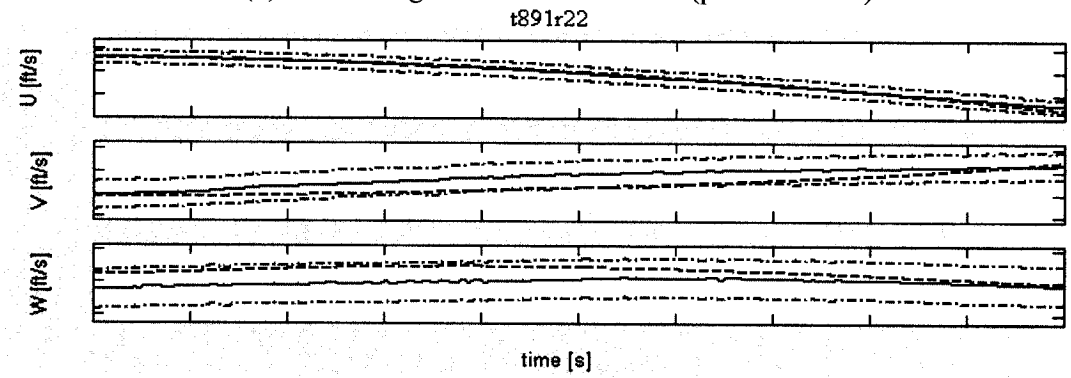
Figure 7-4 Comparison of time history plots of $\alpha, \beta, Mach$



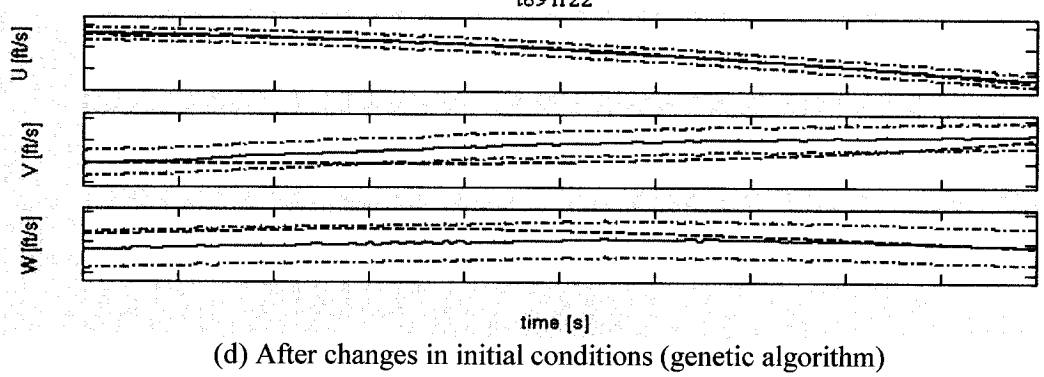
(a) Before changes in initial conditions



(b) After changes in initial conditions (pattern search)

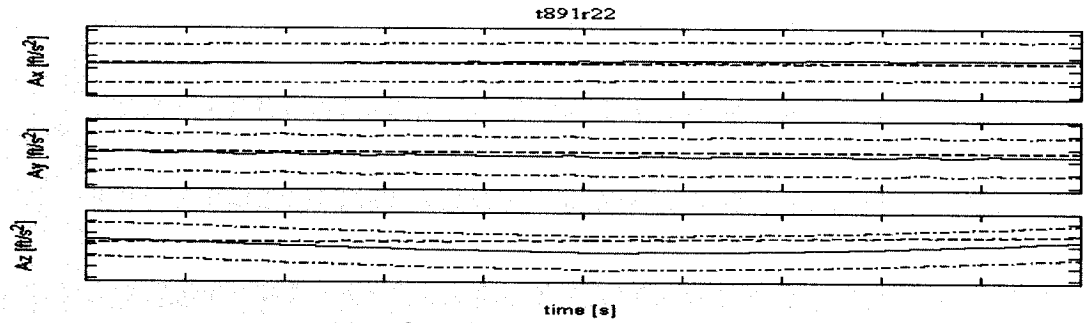


(c) After changes in initial conditions (Nelder and Mead's method)

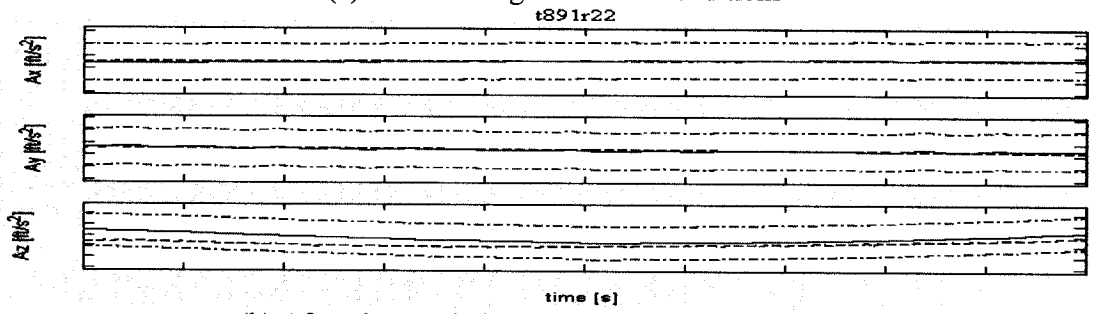


(d) After changes in initial conditions (genetic algorithm)

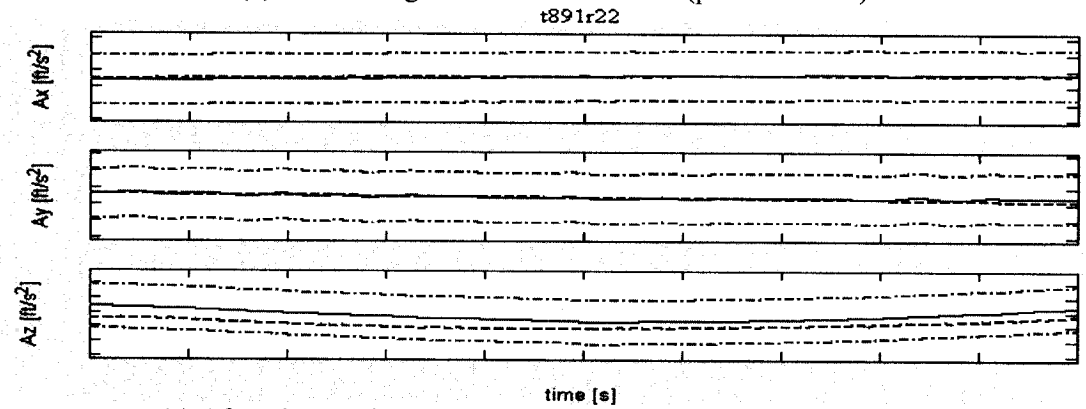
Figure 7-5 Comparison of time history plots of linear velocities u, v, w



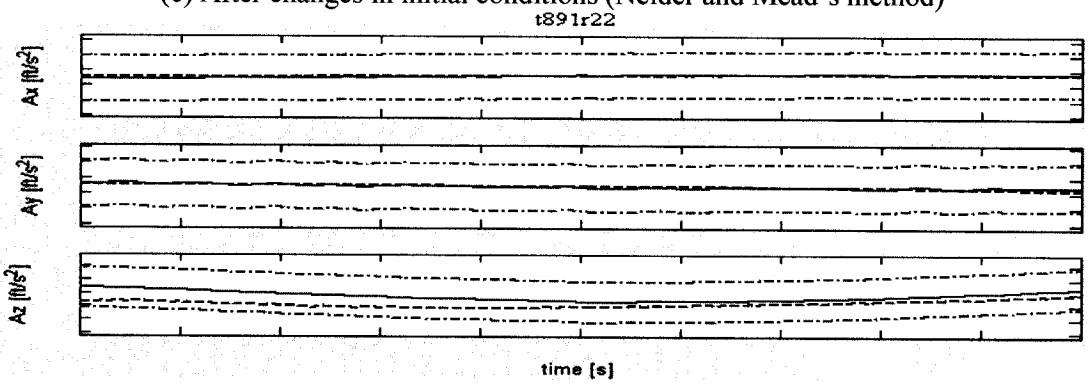
(a) Before changes in initial conditions



(b) After changes in initial conditions (pattern search)

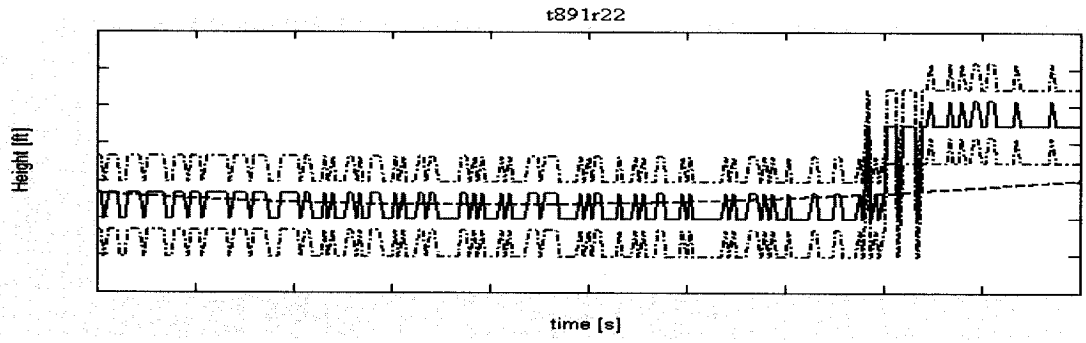


(c) After changes in initial conditions (Nelder and Mead's method)

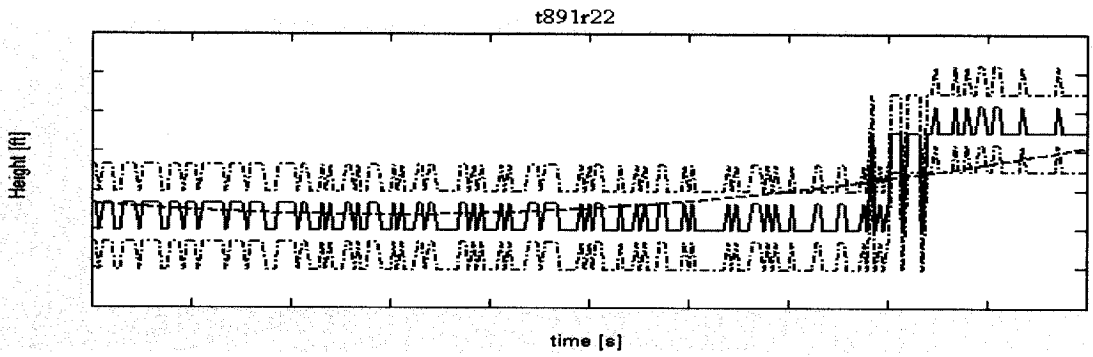


(d) After changes in initial conditions (genetic algorithm)

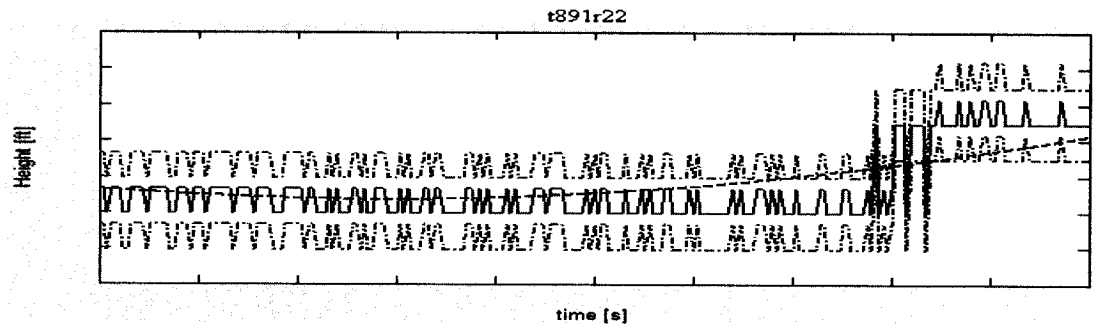
Figure 7-6 Comparison of time history plots of accelerations a_x, a_y, a_z



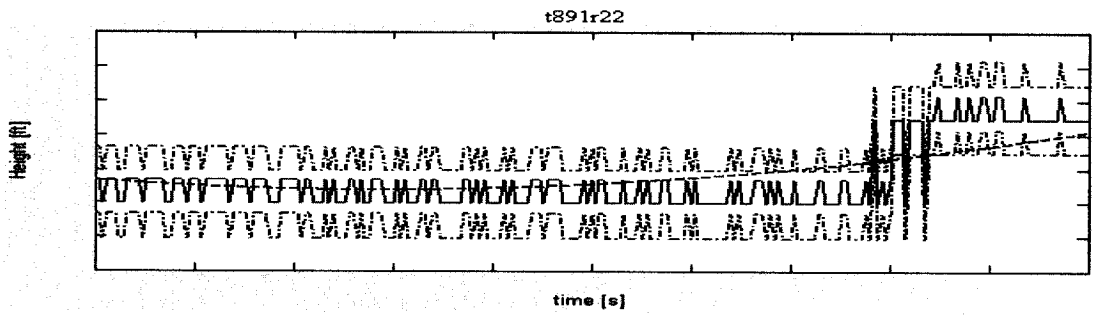
(a) Before changes in initial conditions



(b) After changes in initial conditions (pattern search)



(c) After changes in initial conditions (Nelder and Mead's method)



(d) After changes in initial conditions (genetic algorithm)

Figure 7-7 Comparison of time history plots of flight altitude H

7.2.2 Convergence criteria and speeds of the three algorithms

In numerical computation optimization problems, the convergence speed to the optimum solution is usually the criterion for choosing the best method of computation. The helicopter simulation model takes a time of about 5 seconds to be executed on the computer (AMD Athlon™ XP 1800+, 1.53GHz, 512MB). We therefore estimate that the time for running the simulation model 100 times is about 500 seconds (about 8 minutes). Although the three algorithms are all effective to our optimization problem, we need to try to choose one of the three algorithms We use the running times of the simulation model as the convergence speed criterion of evaluating the three optimization algorithms.

The factors which affect the convergence speed in the three algorithms are presented as follow:

(1) Pattern search

The pattern search convergence speed depends on the starting point and the step-lengths of termination; the convergence criterion is satisfied when the step-lengths are smaller than the pre-determined step lengths of termination. The starting point is arbitrary in the set of solutions, and for this reason we use the convergence steps to control the convergence speeds. In order to increase the convergence speed, we apply different chosen step lengths of termination for different variables. Table 7.2 shows the different step lengths of termination for the decision variables.

Table 7.2 Decision variables and step lengths of termination

Optimization parameters	p	q	r	a_x	a_y	a_z	ϕ
Step length of termination	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Optimization parameters	β	*Mach*	*ROC*	$\dot{p}$	$\dot{q}$	$\dot{r}$	
Step length of termination	0.01	0.0001	5	0.1	0.1	0.1	

(2) Nelder and Mead's method

The convergence speed of the Nelder and Mead's method is based on the standard deviation of the $(n+1)$ function values (equation 6.8) which should be smaller than the pre-determined small value ε; we select $\varepsilon = 5\times10^{-5}$.

(3) Genetic algorithm

The convergence speed of the Genetic Algorithm is related to crossover rate, mutation rate and population size, but no reference gives and introduces the number of more reasonable population. In practice, some researchers choose large population size, but for our optimization problem, using large populations is unreasonable. We choose $popsize = 16$, set the predetermined *generation* = 30, the average predetermined $error \le 5\%$ (for each sub-objective). Here the error expression is determined by equations (5.10) and (5.11) as follows:

$$error = \frac{\sum_{k=1}^{m}\{[x_{measured\ (t_0+k\Delta t)} - x_{simulated\ (t_0+k\Delta t)}]^2 \Delta t^2\}}{\sum_{k=1}^{m}\{[h^{+}_{(t_0+k\Delta t)} - h^{-}_{(t_0+k\Delta t)}]^2 \Delta t^2\}}$$

In section 6.5, we discussed the influence of the mutation rate on convergence speed. We choose different values of mutation rate to observe the convergence speed of our optimization problem. Table 7.3 shows the relationship between mutation rate and the times for running the helicopter simulation model (i.e. convergence speed).

Table 7.3 only shows the results of the six flight case studies with changes in mutation rate, while the other cases show the similar results. We obtain the results as follows:

1. For flight cases t891r21and t892r22, the optimization process is stopped by the predetermined generation because the predetermined error ($\leq 5\%$) is less than the error determined by the helicopter dynamical model.
2. For flight cases t891r23, t891r24, t891r25 and t904r10, when $p_m \leq 0.04$, the convergence speed compared with that when $p_m > 0.04$ is obviously lower, the optimization process sometimes is stopped by the predetermined error ($\leq 5\%$) and sometimes terminated by the predetermined generation. When $p_m > 0.04$, the calculations are always stopped by the predetermined error ($\leq 5\%$). The changes in convergence speed with different p_m values basically appear in the statistical results presented by Figure 6-11.
3. The better value of p_m should be between 0.05 and 0.08, so in GA, we choose the parameters: $generation = 30$, $error \leq 5\%$ (for each sub-objective) $p_m = 0.06$, $p_c = 0.65$, $popsize = 16$

Table 7.3 The relationship between mutation rate and convergence speed

Flight cases	combination	edited variables	Start and end time	Run NO.	Times for running the helicopter simulation model							
					Mutation rate							
					0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08
t891r21	TrimAngAccel subTrimVelRate	p,q,r $\dot{p},\dot{q},\dot{r}$	[6, 11]	1	371	448	458	454	461	477	480	480
				2	384[b]	444	455[b]	460	465	476	480	478
				3	377[b]	437	451[b]	463	473	469	479	480
				4	389	443	457	457	468	480	480	480
				5	382	450[b]	461	459	475	473	480	479
	Average				-	-	-	-	-	-	-	-
t891r22	TrimAngAccel subTrimVelRate	p,q,r $\dot{p},\dot{q},\dot{r}$	[9, 14]	1	373	445	453	461	463	474	480	480
				2	407	439[b]	459[b]	453	474	478	480	479
				3	386[b]	446	457	462	469	480	480	480
				4	392	441	449	466	471	469	478	480
				5	387	444[b]	457	469	459	476	480	480
	Average				-	-	-	-	-	-	-	-
t891r23	TrimAngAccel subTrimRateROC	$p,q,r,\dot{p}$ $\dot{q},\dot{r},ROC$	[6, 11]	1	385	443	456	170*	61*	103*	284*	308*
				2	372	387*	447	145*	103*	106*	152*	312*
				3	296*	363*	404*	337*	202*	125*	207*	92*
				4	388	432	350*	198*	82*	65*	313*	176*
				5	279*	287*	290*	454	44*	145*	70*	103*
	Average				-	-	-	-	***98***	***109***	***125***	***104***
t891r24	TrimAngAccel subTrimRateROC	$p,q,r,\dot{p}$ $\dot{q},\dot{r},ROC$	[8, 13]	1	407	396*	448	237*	235*	328*	386*	334*
				2	101*	173*	107*	109*	73*	64*	83*	247*
				3	235*	447	335*	330*	310*	89*	315*	235*
				4	381	258*	93*	224*	75*	334*	102*	69*
				5	393[b]	276*	413*	313*	424*	271*	235*	107*
	Average				-	-	-	***242***	***223***	***217***	***224***	***198***
t891r25	TrimAngAccel subTrimRateROC	$p,q,r,\dot{p}$ $\dot{q},\dot{r},ROC$	[1, 10]	1	369	447	453	188*	332*	321*	54	141*
				2	315*	358*	120*	313*	67*	103*	361	281*
				3	373	107*	230*	401*	201*	81*	169	133*
				4	445	439	107*	462	89*	169*	103	87*
				5	198*	203*	356*	221*	135*	193*	71	304*
	Average				-	-	-	-	***165***	***173***	***152***	***189***
t904r10	TrimAngAccel subTrimVelRate	p,q,r $\dot{p},\dot{q},\dot{r}$	[1, 6]	1	366[b]	235*	396*	469	403*	307*	378*	312*
				2	298*	98*	126*	235*	77*	158*	106*	81*
				3	106*	451	323*	93*	89*	73*	58*	107*
				4	377	325*	92*	213*	122*	125*	136*	209*
				5	301*	446	267*	218*	196*	173*	344*	73*
	Average				-	-	***241***	-	***177***	***167***	***204***	***156***

1. Optimization Objectives: $p, q, r, \phi, \theta, \psi, \alpha, \beta, Mach, u, v, w, a_x, a_y, a_z, H$
2. The parameters of GA: *generation* = 30, p_c = 0.65, *error* ≤ 5% (each sub-objective), *popsize* = 16
3. The numbers with superscript "*" express the genetic algorithm was stopped by the predetermined *error*, the others express the genetic algorithm was terminated by the *generation*.
4. The numbers with superscript "b" express the optimization results are not good.

7.2.3 Discussion of results

Table 7.4 shows convergence speeds of five flight test cases by using the three algorithms. The other flight cases show similar results. We see that one flight case validation averagely takes less than 30 minutes by use of AutoPOM. We also see that

Table 7.4 The convergence speed of the three algorithms

Flight cases	combination	edited variables	Start and end time	Times for running the helicopter simulation model				
				Pattern search	Nelder and Mead	Genetic algorithm		
						min	max	Average*
t891r21	TrimAngAccel subTrimVelRate	p,q,r $\dot{p},\dot{q},\dot{r}$	[6, 11]	192	122	473	480	475
t891r22	TrimAngAccel subTrimVelRate	p,q,r $\dot{p},\dot{q},\dot{r}$	[9, 14]	205	106	474	480	477
t891r23	TrimAngAccel subTrimRateROC	$p,q,r,\dot{p}$ $\dot{q},\dot{r},ROC$	[6, 11]	153	69	65	145	109
t891r24	TrimAngAccel subTrimRateROC	$p,q,r,\dot{p}$ $\dot{q},\dot{r},ROC$	[8, 13]	203	97	64	334	217
t891r25	TrimAngAccel subTrimRateROC	$p,q,r,\dot{p}$ $\dot{q},\dot{r},ROC$	[1, 10]	121	95	81	321	173
t904r10	TrimAngAccel subTrimVelRate	p,q,r $\dot{p},\dot{q},\dot{r}$	[1, 6]	142	87	73	307	167
Average times				169	96	204	344	296
Average time (minute)				14.1	8	17	28	22

1. Computer: AMD AthlonTM XP 1800+, 1.53GHz, 512MB, one time for running the helicopter simulation model takes 5 seconds
2. Optimization Objectives: $p, q, r, \phi, \theta, \psi, \alpha, \beta, Mach, u, v, w, a_x, a_y, a_z, H$
3. *error* $\leq$ 5% (each sub-objective), *popsize* = 16, *generation* = 30, p_m=0.06, p_c = 0.65

*Average is the statistic of five test times.

the convergence speed of Nelder and Mead's method seems faster than that of pattern search. Pattern search starts from one point and consists of a sequence of exploration steps; some of them are failure steps which delay the convergence speed. Nelder and Mead's method starts from $(n+1)$ points also like pattern search to replace the worst

value by a new better value, it uses more information from $(n+1)$ points and each step is toward the convergence zoom. GA convergence speed varies due to random choice of the crossover sites and mutation sites. Crossovers and mutations may cause the loss of better genes, i.e. the offspring may be worse than their parents, which reduce the convergence speed. Therefore, in the GA, the convergence speed can be either fast or slow as shown in Table 5.4. Although the fastest GA convergence speed is faster than pattern search and Nelder and Mead's method, the average speed is basically slower than Nelder and Mead's method. The maximum and minimum convergence speeds of the flight cases t891r21and t891r22 are very close because when the predetermined error ($\leq$5%) cannot be met, then the calculation does not terminate until the predetermined $generation = 30$, in this case, GA takes more time than Nelder and Mead's method and Hooke and Jeeves' method.

From Table 7.4 and the analysis above, we see that Nelder and Mead's method should be our first choice because it has the fastest convergence speed.

CONCLUSIONS

The validation of an aircraft (here a helicopter) dynamics simulation model involves the evaluation and approval of the model, in accordance with preset validation requirements by use of flight test data. The global model of the Bell 427 helicopter has passed through the QTG list and has been compared with the flight tests. One of the best simulations was picked to demonstrate the matching of the simulation with the flight tests as required by the FAA AC 120-63 [15]. From the analysis of flight cases run with the latest version 7.2 of POM, the global model output plots were shown to be within the tolerances present in reference [15] by changing the initial conditions.

The simulation model validation process is time consuming by use of the software POM. Usually, the validation of one flight test case takes the POM user about 4-6 hours. Fortunately, the validation process can be defined as a mathematical multi-objective optimization problem. We set up the objective function which is a utility function of Least Square cost functions with their weights. The objective function is expressed by equation (5.14). This type of objective functions is convex and unimodal in the optimization domain.

Our optimization problem can not be solved by use of direct search methods. It can be done by use of indirect search methods and genetic algorithm (GA), because the decision vector differs from the objective function vector. We can not find the expressions for the first partial derivatives of the objective function with respect to the

decision vector. In this thesis, we take three algorithms: Hooke and Jeeves' method (pattern search), Nelder and Mead's method, genetic algorithm.

In order to reduce time consumption of the flight case validation, we apply the three algorithms to the software POM and develop the software AutoPOM. AutoPOM runs in Matlab environment as POM does, its interface is created by use of Matlab Graphical User Interface (GUI) technique. This interface is very user friendly as shown in Figure 7-1. Usually, in the computer (AMD Athlon™ XP 1800+, 1.53GHz, 512MB), the validation of one flight test case takes the user about 4-6 hours with POM. The newly developed software AutoPOM reduces the computation time to 20-30 minutes (see Table 7.4). This is 12 times faster than using POM. This translates into a huge time saving when a large number of test cases (e.g. for the Bell 427 simulator certificate program, about 550 flight test cases) is to be analyzed. AutoPOM is also hands-off and the results are perfectly repeatable since no trial and error is involved.

In AutoPOM, we provide the three optimization algorithms for the user to choose from. All of the three algorithms, proved by flight cases, are able to quickly obtain better validation results. The convergence speeds of the three algorithms are different. Pattern search starts from one point, but its failed exploration steps delay the convergence speed. Nelder and Mead's method starts from $(n+1)$ points, it uses more information from $(n+1)$ points and each step moves forward toward the convergence zoom. GA convergence speed varies due to randomly choice of the crossover sites and the mutation sites. The offspring may be worse than their parents, the convergence speed can be either fast or slow as shown in Table 7.3 and 7.4. When the predetermined error can not

be met, GA is stopped by predetermined generation. In this situation, GA usually takes more time than Nelder and Mead's method and Hooke and Jeeves' method. Experimental results show that Nelder and Mead's method has the fastest convergence speed out of the three algorithms, so it is our first chosen method.

The software AutoPOM was presented to Bell Helicopter Textron as a part of the CRIAQ project 3.4 work.

Reference

[1] A.R.S. Bramwell, Helicopter Dynamics, New York: John Wiley & Sons, Inc. 1976.

[2] Gareth D. Padfield, Helicopter Flight Dynamics, The Theory and Application of Flying Qualities and Simulation Modeling, American Institute of Aeronautics and astronautics, Inc. 1996.

[3] G. Arfken, Mathematical Methods for Physicists, 3rd. ed., New York: Academic Press, 1985.

[4] K. Schmidt-Rohr, H.W. Spiess, Multidimensional Solid-State NMR and Polymers, London, Academic Press, 1994.

[5] W.P. Power, R.E. Wasylishen, S. Mooibroek, B.A. Pettitt and W. Danchura, Simulation of NMR Lineshapes Of Quadrupolar Nuclei with Half-Integer spin at Low Symmetry Sites.Journal of Physical and Chemical, Vol. 94, pp 591-598, 1990.

[6] M.E. Rose, Elementary Theory of Angular Momentum, New York: John Wiley & Sons, 1957.

[7] Brian L. Stevens and Frank L. Lewis, Aircraft control and Simulation, John Wiley & Sons, 1992.

[8] John H. Blakelock, Automatic Control of Aircraft and Missiles, New York: John Wiley & Sons, Inc. 1987.

[9] Raymond W. Prouty, Helicopter Performance, Stability and Control, Florida: Robert E. Krieger Publishing Company, 1990.

[10] J.R. RAOL, L. SARASWATHI, and G. GIRIJA, Helicopter Flight Test Specifications for Parameter Estimation Flight Mechanics & Control Division, Project Document FC 9313 India August 1993

[11] Richard S. Shevell, Fundamentals of Flight, U S Prentice-hall, Inc. 1989.

[12] Alastair K. Cooke and Eric W.H. Fitzpatrick, Helicopter Test and Evaluation, American Institute of Aeronautics and astronautics, Inc. 2002.

[13] Wolowicz C. H., Considerations in the determination of stability and control derivatives and dynamic characteristics from flight data, AGARD Rep.549-Part 1, 1966.

[14] Andrei Vladimir Popov, Proof-of-Match technique for Bell 427 helicopter level D simulator, M.S thesis, École de Technologie Supérieure Univesité de Quebéc, October, 2005.

[15] Federal Aviation Administration, Advisory Circular AC 120-63 (1994) Helicopter simulation qualification, U.S Department of Transportation.

[16] M.Jenkinson and S.M. Smith, A global optimization method for robust affine registration of brain images, Medical Image Analysis, vol. 5, No. 2, pp. 143--156, 2001.

[17] R.T. Marler and J.S. Arora, Survey of multi-objective optimization methods for engineering, Structural and Multidisciplinary Optimization, vol. 26, pp.369-395, 2004.

[18] V. Chankong, and Y.Y. Haimes, Multiobjective Decision Making Theory and Methodology, New York: Elsevier Science Publishing, 1983.

[19] T.L. Vincent and W.J Grantham, Optimality in Parametric Systems, New York: John Wiley and Sons Inc. 1981.

[20] Leon Cooper and David Steinberg, Introduction to Methods of Optimization, Toronto: W.B. Saudners Company 1970.

[21] Boris T. Polyak, Introduction to Optimization, New York: Optimization Software, Inc. 1987.

[22] Douglass J. Wilde and Charles S. Beightler, Foundation of Optimization. Englewood Cliffs: N.J, Prentice-Hall, Inc. 1967.

[23] Genetic Algorithms in Engineering Systems, Edited by A. M.S. Zalzala and P.J. Fleming, London: The Institution of Electrical Engineers.

[24] Mitsuo Gen and Runwei Cheng, Genetic Algorithm and Engineering Design, New York: John Wiley and Sons Inc 1997.

[25] S.J. Wu and P.T. Chow, Steady-state Genetic Algorithm for Discrete Optimization of Trusses, Computers and Structures Vol.56, No.6, pp979-991, 1995.

[26] C.L. Karr, D.A. Stanley and B.J. Scheiner, Genetic Algorithm Applied To Least Squares Curve Fitting, Report of investigations 9339. U.S. Department of Interior Bureau of Mines 1991.

[27] A.T.C.Goh, Genetic Algorithm Search for Critical Slip Surface in Multiple-wedge Stability Analysis, NRC. Canadian Geotechnical Journal Vol. 36, 1999.

[28] Zbigniew Michalewicz, Genetic Algoriththm +Data Structures=Evolution Program, 3rd ed. New York: Springer-Verlag, 1996.

[29] M.J. Box, D. Davies and W.H. Swann, Non-linear Optimization Techniques, Monograph No. 5, England: ICI Ltd., 1969.